International Society for Photogrammetry and Remote Sensing
Société Internationale de Photogrammétrie et de Télédétection
Internationale Gesellschaft für Photogrammetrie und Fernerkundung

THE INTERNATIONAL ARCHIVES OF THE PHOTOGRAMMETRY, REMOTE SENSING AND SPATIAL INFORMATION SCIENCES
ARCHIVES INTERNATIONALES DES SCIENCES DE LA PHOTOGRAMMÉTRIE, DE LA TÉLÉDÉTECTION ET DE L'INFORMATION SPATIALE
INTERNATIONALES ARCHIV FÜR PHOTOGRAMMETRIE, FERNERKUNDUNG UND RAUMBEZOGENE INFORMATIONSWISSENSCHAFTEN

VOLUME / VOLUME / BAND XXXVIII PART / TOME / TEIL 3 / W4

CMRT09

Object Extraction for 3D **C**ity **M**odels, **R**oad Databases and **T**raffic Monitoring - Concepts, Algorithms and Evaluation

Paris, France
September 3 – 4, 2009

Papers accepted on the basis of peer-reviewed full manuscripts

Editors
U. Stilla, F. Rottensteiner, N. Paparoditis

Organised by
ISPRS WG III/4 – Complex scene analysis and 3D reconstruction
ISPRS WG III/5 – Image sequence analysis
MATIS Laboratory, Institut Géographique National, Saint-Mandé, France
Société Française de Photogrammétrie et de Télédétection (SFPT)

Supported by
Photogrammetry and Remote Sensing, Technische Universitaet Muenchen (TUM)
Institute of Photogrammetry and GeoInformation, Leibniz Universitaet Hannover

ISSN 1682-1750

Published by
- ISPRS WG III/4 – Complex scene analysis and 3D reconstruction
- ISPRS WG III/5 – Image sequence analysis

ISPRS Headquarters 2008-2012
c/o CHEN JUN, ISPRS Secretary General
National Geomatics Centre of China
No. 1 Baishengcun, Zizhuyuan
Beijing 100048, PR CHINA
Tel: +86 10 6842 4072
Fax: +86 10 6842 4101
Email: chenjun@nsdi.gov.cn; chenjun_isprs@263.net
ISPRS WEB Homepage: http://www.isprs.org

Available from
GITC bv
P.O.Box 112
8530 AC Lemmer
The Netherlands
Tel: +31 (0) 514 56 18 54
Fax: +31 (0) 514 56 38 98
E-mail: mailbox@gitc.nl
Website: www.gitc.nl

Workshop Committees

Conference Chairs:

Uwe Stilla (Technische Universität München)
Franz Rottensteiner (Leibniz Universität Hannover)
Nicolas Paparoditis (Institut Géographique National)

Cooperating Working Groups:

WG III/4 – Complex scene analysis and 3D reconstruction
WG III/5 – Image sequence analysis

Local Organizing Committee:

Caroline Baillard (SIRADEL, Rennes)
Francois Boyero (Institut Géographique National)
Frédéric Bretar (Institut Géographique National)
Adrien Chauve (Institut Géographique National)
Isabelle Grujard (SFPT)
Ludwig Hoegner (Technische Universität München)
Clément Mallet (Institut Géographique National)
Nicolas Paparoditis (Institut Géographique National)
Marc Pierrot-Deseilligny (Institut Géographique National)
Laurent Polidori (SFPT)
Olivier Tournaire (Institut Géographique National)
Wei Yao (Technische Universität München)

Program Committee:

Selim Aksoy (Bilkent University)
Caroline Baillard (SIRADEL, Rennes)
Emmanuel Baltsavias (ETH Zurich)
Claus Brenner (Leibniz Universität Hannover)
Mathieu Brédif (IGN / MATIS, St. Mandé)
Matthias Butenuth (Technische Universität München)
Nicolas Champion (IGN / MATIS, St. Mandé)
Matthieu Cord (UPMC PARIS-6)
Wolfgang Förstner (University of Bonn)
Paolo Gamba (University of Paia)
Markus Gerke (ITC)
Norbert Haala (Universität Stuttgart)
Stefan Hinz (Universität Karlsruhe)
Thomas Kolbe (Technische Universität Berlin)
Simon Lacroix (LAAS/CNRS)
Maxime Lhuillier (LASMEA UMR 6602, UBP/CNRS)
Helmut Mayer (Universität der Bundeswehr München)
Chris McGlone (SAIC, Chantilly, VA)
Franz Meyer (Alaska Satellite Facility, Fairbanks)
Sönke Müller (Leibniz Universität Hannover)
Nicolas Paparoditis (IGN / MATIS, St. Mandé)
Norbert Pfeifer (Vienna University of Technology)
Ralf Reulke (Humboldt-Universität zu Berlin)
Franz Rottensteiner (Leibniz Universität Hannover)
Jie Shan (Purdue University)
Uwe Soergel (Leibniz Universität Hannover)
Gunho Sohn (York University)
Uwe Stilla (Technische Universität München)
Olivier Tournaire (IGN / MATIS, St. Mandé)
Florence Tupin (Telecom ParisTech)
Bruno Vallet (IGN / MATIS, St. Mandé)
Thomas Voegtle (Universität Karlsruhe)

Preface

Automated extraction of topographic objects from remotely sensed data is an important topic of research in Photogrammetry, Remote Sensing, GIS, and Computer Vision. This joint conference of ISPRS working groups III/4 and III/5, held in Paris, France, discussed recent developments, the potential of various data sources, and future trends both with respect to sensors and processing techniques in automatic object extraction. The focus of the conference lay on methodological research. It was held in conjunction with ISPRS Laser-scanning conference.

The conference addressed researchers and practitioners from universities, research institutes, industry, government organizations, and private companies. The range of topics covered by the conference is reflected by the terms of reference of the cooperating ISPRS working groups:

- Complex Scene Analysis and 3D Reconstruction (WG III/4)
- Image Sequence Analysis (WG III/5)

Prospective authors were invited to submit full papers of a maximum length of 6 pages. We received 60 full papers for review. The submitted papers were subject to a rigorous double blind peer review process of full papers. Altogether 38 papers were accepted based on the reviews. This corresponds to a rejection rate of 37%. Each paper was reviewed at least by two members of the program committee. The accepted papers and one invited paper were published as printed proceedings in the IAPRS series as well as on CD-ROM. Only a subset of these papers could be presented orally due to the single track design of CMRT09 and the generous time slots for intensive discussion.

In total, we received contributions from authors coming from 20 countries. The proceedings include 39 papers from authors coming from 14 countries. There were 7 oral sessions with altogether 23 papers and one interactive session where 16 papers were presented as posters.

Finally, the editors wish to thank all contributing authors and the members of the Program Committee. In addition, we like to express our thanks to the Local Organising Committee, without whom this event could not have taken place. Ludwig Hoegner did a great job with the management of the conference tool. The final word processing of all incoming manuscripts and the preparation of the proceedings by Wei Yao are gratefully acknowledged. Olivier Tournaire did a great job with the CD-ROM edition of the proceedings, and so did Clement Mallet and Adrien Chauve with the registration and the choice of the gala dinner. We would also like to thank Clement Mallet, Adrien Chauve, Frederic Bretar, Marc Pierrot-Deseilligny, Olivier Tournaire, Isabelle Grujard, François Boyero, Carol Godin, and Jessica Vencatasamy for the general day-to-day organisation of the event, and at last Nicolas Paparoditis for managing the Local Organising Committee.

Munich, Hannover and Paris, July 2009

Uwe Stilla

Franz Rottensteiner

Nicolas Paparoditis

Contents

EFFICIENT ROAD MAPPING VIA INTERACTIVE IMAGE SEGMENTATION

O. Barinova, R. Shapovalov, S. Sudakov, A. Velizhev, A. Konushin

Moscow State University, Dept. of Computational Mathematics and Cybernetics
{obarinova, shapovalov, ssudakov, avelizhev, ktosh}@graphics.cs.msu.ru

Commission III, WG III/5

KEY WORDS: Automation, Video, Processing, Incremental, Learning, Object, Detection

ABSTRACT:

Last years witnessed the growth of demand for road monitoring systems based on image or video analysis. These systems usually consist of a survey vehicle equipped with photo and video cameras, laser scanners and other instruments. Sensors mounted on the van collect different types of data while the vehicle goes along the road. Recorded video can be geographically referenced with the help of global positioning systems. Road monitoring systems require special software for data processing. This paper addresses the problem of video analysis automation, and particularly the pavement monitoring functionality of such mobile laboratories. We show that computer vision methods applied to this problem help to reduce amount of manual labour during data analysis. Our method transforms video collected by mobile laboratory into rectified geo-referenced images of road pavement surface, and allows mapping of lane marking and road pavement defects with minimum user interaction. In our work the mapping workflow consists of two stages: off-line and online stage. In order to reduce user effort during error correction we take advantage of hierarchical image segmentation, which helps to delete false detections or mark missing objects with just a few clicks. Through continuous training of detection algorithm with the help of operator input error rate of automatic detection decreases; thus minimal input is required for accurate mapping. Experiments on real-world road data show effectiveness of our approach.

1. INTRODUCTION

Roadway monitoring systems are widely-used for supervising road pavement surface and repair planning. These systems usually include a complex of video cameras and other sensors mounted on a car as shown on Figure 1. The sensors record road pavement surface when travelling on a pavement at traffic speed.

Most existing software for road monitoring involves manual processing of video collected by these mobile laboratories. Operator manually marks objects like lane marking and pavement surface defects (potholes, cracking and patches) on each video frame. This procedure is laborious and takes plenty of time; therefore the task of automation of objects detection comes into focus. In this paper we consider the problem of automation of video analysis for pavement surface monitoring. We describe a tool which assists in utilising visual observation data of pavement surface and mapping lane marking and pavement surface defects.

Our main goal is to minimize effort of operator at the time of mapping lane marking and road defects while preserving accuracy of mapping result. The effectiveness of our method is achieved by intensive usage of computer vision techniques together with user-friendly interface that allows checking results of automatic detection and correcting errors if needed.

As long as direct mapping of lane marking and road pavement defects in video sequences faces severe difficulties, we transform video into rectified images of road pavement surface. These images are further processed during interactive mapping.

While to our knowledge there haven't been much research on topic of road defects detection, lane detection is a well-researched area of computer vision with applications in autonomous vehicles and driver support systems. Despite perceived simplicity of finding white markings on a dark road, it can be very difficult to determine lane markings on various types of road. These difficulties arise from shadows, changes in the road surfaces itself, and differing types of lane markings. A lane detection system must be able to pick out all manner of markings from cluttered roadways and filter them to produce a reliable estimate of the vehicle position and trajectory relative to the lane as well as the parameters of the lane itself such as its curvature and width.

Existing methods for lane marking detection are usually based on edge detection (McDonald, 2001) and gradient analysis (Lu, 2007). Use of edges makes detection results sensitive to noise, changes in lighting conditions and shadows. Another approach uses steerable filters (McCall, 2004) which can be convolved with the input image and provide features that allow them to be used to detect both dots and solid lines while providing robustness to cluttering and lighting changes.

As long as these methods were designed for autonomous vehicles, they aim at tracking of lane marking in video. In our work the goal is to detect lane marking in still images of road surface. Also our task is to detect precise contours of lane marking instead of just determining lane marking direction. This task is closely related to the field of semantic image segmentation, therefore the method we propose for detection is based on semantic segmentation of rectified road images.

Rectified images can differ substantially depending of roadway material, time of survey and weather conditions. Therefore automatic detection tuned on one road image can perform poorly on other images. For this reason we have developed a detection algorithm which is automatically tuned with the aid of user interaction in order to perform best on each particular road. This allows accounting for specific characteristics of every particular road, or even a road section.

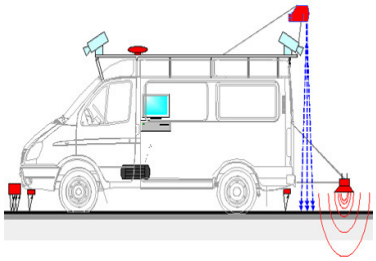
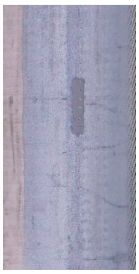


Figure 1. Road laboratory. Video cameras are mounted on the front side and on the back side of the car.

(a)



(b)

Figure 2. Video frame from one camera and a corresponding section of rectified road image.

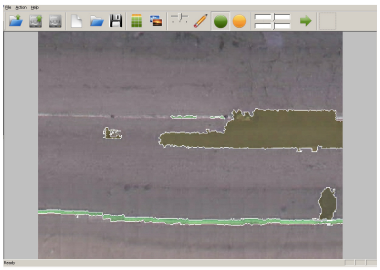


Figure 3. Lane marking and road defects are mapped with a minor user interaction.

The outline of mapping process for a user if the following: first automatic detection is applied to a small section of rectified road image, after that user checks the results of automatic method, corrects the errors if needed and then detection algorithm is adapted in order to take new data into account. After that user goes on to the following road section and the whole procedure is repeated again. Through continuous training of detection algorithm with the help of operator input error rate of automatic detection decreases; thus minimal input is required for accurate mapping. In order to reduce user effort during error correction we take advantage of hierarchical image segmentation, which helps to remove false detections or mark missing objects with just a few clicks.

The paper is organized as follows. Section 2 addresses the procedure of data acquisition and transformation of video sequences into rectified road images. Section 3 describes offline stage of our method. Section 4 gives details on user interaction with the system. Our method for lane marking and pavement surface defects detection is described in section 5. Section 6 is devoted to our machine learning algorithm, which helps to tune detectors on various road images individually. Experiments on real-world data collected by our mobile laboratory are described in section 7. Section 8 is left for conclusion and future work.

2. DATA ACQUISITION

In this work we have used a vehicle equipped with 4 video cameras with resolution 720x576px and Global Positioning System (GPS) on board. The cameras capture video of road surface and roadside, which can be accurately geographically registered by means of GPS. Figure 2 (a) shows an example of one frame of video obtained by a video camera mounted on a van and corresponding section of rectified road image.

Although all cameras in capture video, usage of video as input for mapping lane marking and road pavement defects has severe drawbacks. First, areal objects on road pavement surface suffer from projective distortion which degrades performance of detection algorithms. For example, rectangular pavement patches become trapezoids in video frame. Second, some elongated objects are not fully visible in any single frame of video sequence. Third, different objects are represented with different spatial resolution on the same video frame depending on their distance to the camera. To overcome these problems we transform video sequence into rectified image of the road pavement surface.

These images are obtained from video using perspective plane transformation. Resulting image is one long image in the full driven length. All rectified images are stored in raw format with time and distance information of all pixels. Figure 2 (b) shows an example of video frame obtained from one camera and a corresponding section of rectified image of road pavement surface.

As long as image processing algorithms (like image segmentation) used at subsequent stages of our workflow are memory and time consuming, long rectified road image are cut into non-overlapping small sections. Each part is about 0.5 megapixel image and represents an approximately 5-10 meters long section of road pavement surface. All these section images are further processed in chain, following vehicle path.

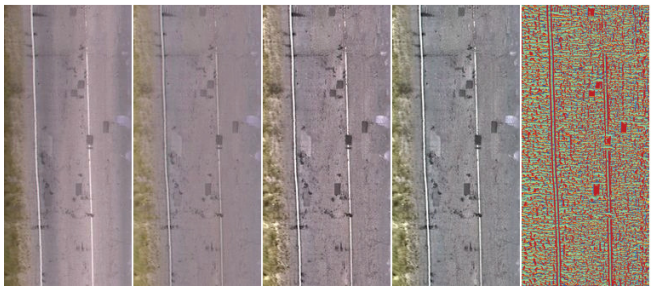
3. OFFLINE STAGE OF MAPPING PROCESS

In our work the mapping workflow consists of two stages: off-line and online stage. As long as we aim at interactive working time at the time of road mapping, all time-consuming operations required by both detection and learning are performed off-line. Offline stage happens once for each road data before user starts mapping road surface. This stage doesn't require any user assistance. Our detection algorithm is based on over-segmentation and classification of super-pixels, therefore offline stage includes image processing, image segmentation, and calculation of features for each image segment. Below these operations are described in more details.

Image processing

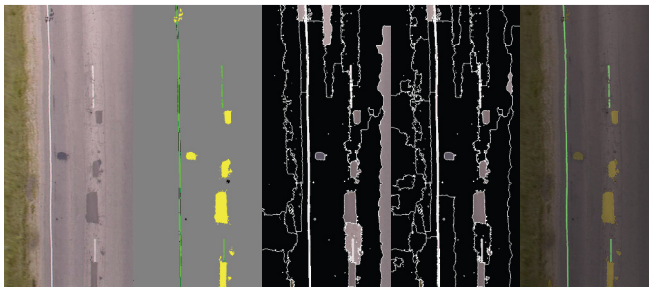
Roadway images are strongly differed to each other in color, brightness and texture. This fact substantially complicates the detection task. Therefore main goal of image preprocessing is to normalize images and put them into some standard state. Image processing includes luminance correction, contrast adjustment, colour correction and image smoothing. All these operations are performed in CIE-Lab colour space.

For luminance correction we use a modification of Retinex algorithm(Land, 1971) . Single-Scale Retinex has artifacts such as halos around dark objects and shadows around light ones, what damages detection in low-contrast images. Conventional Multi-Scale Retinex also has these artifacts when it has to deal with strong luminance changes. Since most of necessary lightness correction is caused by ruts on the road, brightness map is calculated using elongated median filter. It helps to reduce halos effect during luminance corrections (Figure 4 (b)).



(a) (b) (c) (d) (e)

Figure 4. Image processing stages: (a) Source image; (b) Retinex transformation result; (c) result of retinex and contrast adjustment stages; (d) overall image preprocessing result; (e) Texton map



(a) (b) (c) (d) (e)

Figure 5.Cascade classification stages. (a) – Input image, (b) – ground truth image, (c) – 1st cascade layer result, (d) – 2nd cascade layer result, (e) – overall algorithm result.

After Retinex correction mean value of luminance becomes equal to one. Before scaling L-component back to normal, contrast adjustment is achieved by squaring it. Then, L-channel is scaled back to normal (Figure 43(c)). This operation helps to make detection of road defects easier even in low-contrast images.

Colour correction uses conventional grey world algorithm. In Lab colour space it consists of the shift of colour components so as to make mean value of these components be equal to zero, instead of scaling colour components in RGB space. In the final stage bilateral filtration is used to smooth image without loss of important details (Figure 4(d)).

Image segmentation

The hierarchical structures is a powerful tool to analyze data in many applications. Several basic approaches to construction of such multi-level image structure exist. The first approach involves recursive segmentation. An image is segmented in a large scale, and then segments are independently split into pieces. Another approach involves successive segmentation of an image at several scales. But in this case large segments not necessarily represent combinations of smaller ones; this fact limits the scope of application of this method for segmentation.

In this work we used a method based on determination of strength of the boundaries between segments by means of the analysis of saddle points between density modes and merging segments that are weakly separated. For segmentation of the image in our work the hierarchical version of algorithm of mean shift, proposed in (Paris, 2007) is used.

This algorithm provides fast hierarchical segmentation on the basis of idea of the saddle point analysis. Results of this hierarchical segmentation are shown in Figure 5, where borders of segments at different levels of hierarchy are shown in white.

Features calculation

A number of various features are used for classification of segments. We use colour statistics, such as mean values of CIE Lab components and mean values of RGB components, colour variance, Lab components' percentiles.

To account for shape information we calculate coordinate statistics, such as mass centre, coordinate variance, elongation, orientation, area of the segment. Usage of information about neighbourhood of the segment is also very informative for road defects detection. Accordingly distance between mean values of colour components inside segment and inside its neighbourhood are also included in the list of features.

Texton histograms are also used in our system (Leung 1999). These features are proven to be highly effective in recognition task and are used nowadays in many detection and recognition systems (Criminisi, 2006).

Previously created filter bank is applied to the image; filter output vectors for each pixel are associated with the nearest texton vectors from previously trained universal texton dictionary. Then histogram of textons over the segment is used as feature for classification task. Figure 4(e) illustrates a resulting texton map, which is an image, where pixels are labeled accordingly to corresponding textons.

4. ONLINE STAGE OF MAPPING PROCESS

At online stage automatic detection algorithm is applied to parts of rectified road image. User examines results of automatic detection on one image part and corrects detection errors if needed. Then automatic detection is adapted to new data. After that user goes on to the next part of the road and again analyses and corrects results of automatic detection. Accordingly automatic detector is continuously tuned in order to capture specifics of particular road.

Our system provides various facilities for making process of error correction easier for the user. The GUI contains a control which lets user change segmentation level. Operator is able to mark ground truth in a less detailed level and then specify it in a more detailed one. It makes user work more efficient.

Another facility allows controlling tradeoff between detection rate and false positive rate individually for lane marking and road defects. For example, user can increase detection rate of road defects detection (thus increasing false positive rate) by moving a slider. The change in detection rate is performed by changing a threshold on classifier output for road defects on the last cascade layer. This feature helps to significantly reduce amount of manual work in the beginning of online stage, when classifiers show instable performance.

5. LANE MARKING AND PAVEMENT DEFECTS DETECTION ALGORITHM

Our approach is based on cascade classifiers. The idea of cascades is derived from (Viola, 2002). General workflow of cascades is the following. There is ordered set of classifiers, where every subsequent classifier is more "complex" than the preceding one ("complexity" of the classifiers is defined depending on specifics of data or application). Input data

array is passed through these classifiers in turn; each classifier eliminates the data that confidently does not belong to the target class, the remained data is passed to the following, more "complex" classifier, for more thorough examination, etc.
The general idea of cascades involves detection of one target class that implies binary classification. In our task the cascade is applied to a problem of separating objects of two different classes from a background; that requires three-class classification. It is important to notice, that the background class in our task dominates significantly over classes of lane marking and pavement defects. This finding suggests modifying the scheme of cascades used in (Sudakov, 2008) in order to allow detection of several classes of objects.

Cascade workflow

At the offline stage the image had been segmented using the method from (Paris, 2007) into homogeneous regions, and several scales of segmentation are available. The largest scale segmentation is used on the first layer of cascade, the most detailed segmentation scale is used on the last layer. Segmentation at each subsequent scale is a subdivision of segmentation at the preceding scale, therefore we have a sequence of enclosed segments (hierarchy). Each cascade layer corresponds to a certain scale of hierarchy of segmentation and a binary classifier.
Those segments that have not been rejected at the preceding layers of cascade are classified into two classes: objects of interest (including lane marking and road surface defects) and background. The goal of classification is to reject the segments that do not contain pixels of objects of interest. For this purpose the threshold on the classifier output is set up so that the detection rate is close to 100 %.
This procedure is repeated up to the last cascade layer and then multi-class classification is applied. Segmentation corresponding to the last cascade layer is detailed enough to capture precise bounds of lane marking and pavement defects. Moreover, the majority of background segments are rejected at the preceding layers, so the number of background segments passed to the last layer approximately equals to the number of lane marking segments and segments of road covering defects. Therefore our cascade operational scheme also helps to solve a problem of imbalanced classes thus helping to achieve better classification performance. The workflow of cascaded segmentation is illustrated in Figure 5.

6. ON-LINE LEARNING

Online learning algorithms (Domingos, 2000, Oza,2005) process each training example once ‘on arrival’ without the need for storage and reprocessing, and maintain a current model that reflects all the training examples seen so far. Such algorithms have advantages over typical batch algorithms in situations where data arrive continuously. They are also useful with very large data sets on secondary storage, for which the multiple passes through the training set required by most batch algorithms are prohibitively expensive.
In order to enable user-aided tuning of object detection we incorporated on-line learning algorithm in the core of the system. As long as we aim at interactive time of classification and learning, the following requirements for the online-learning algorithm arise. First, online classifier should not store previously seen training examples. Second, learning time should not depend on the number of examples already seen by the learner. Thus we chose online random forest over Hoeffding trees (Domingos, 2000) as it meets both these requirements. Below we describe how online classifiers are used in our cascaded detection method.

On-line learning of cascaded segmentation

In section 3 we described the workflow of cascaded algorithm for object detection, supposing that all classifiers are already trained. Here we describe the training phase of cascaded detection method.
The main problem here is what data should be used for training of classifier at each particular layer of cascade. There are two difficulties with providing training data to classifiers at cascade layers. First, we should take into account all segments which contain target class because if we do not provide enough samples of target class at the training stage, classifier wouldn’t be able to detect them at classification stage. This can lead to severe error of first kind.
The second problem is lack of target samples at all cascade layers in comparison to number of background samples. This class imbalance can lead to additional increase of error of first kind. This means that cascade will miss large amount of target objects. Therefore we need to consider special techniques for balancing class distributions. Our solution for both these problems is the following. We train classifier corresponding to each cascade layer using the data passed to a corresponding cascade layer by preceding version of cascade which had not been adapted to last portion of data. Then, all segments which contain marking and defects are added to the training set on each layer. In order to better balance classes’ distribution we use cost-sensitive online random forest, described below.

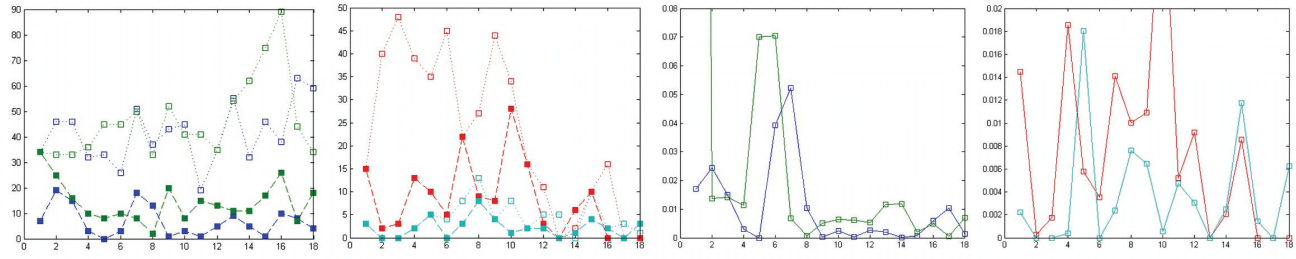
On-line random forest

In this work we use ‘one vs all’ algorithm for multi-class classification on the last cascade layer. This enables using binary classifiers on the lowest tier of the system. Those classifiers should be able to learn even some first portions of a training set efficiently to give a reliable classification result. Also, as mentioned above, these classifiers should handle imbalanced classes’ data.
We use on-line random forest classifiers at all stages of cascade. Our version of online random forest resembles an online bagging algorithm proposed by (Oza, 2005). We modified this algorithm in order to allow balancing the classes. This is achieved by assigning parameters of exponential distribution individually to each class in on-line bagging algorithm.
This procedure akin to random resampling is equivalent to introducing different penalty costs for misclassification of objects of each class. In this work costs are calculated in inverse proportion to a number of samples in the class. Also we use a random set of features for every weak classifier like in Random Forest algorithm (Brieman, 2001). This, together with using Hoeffding trees (Domingos, 2000) as a weak learner, helps to achieve stable classification results and reduce training and classification time.

7. - EXPERIMENTS

Image base

For the experiments we used four road images. They differ in quality and marking and relative areas of defects. We tested our system on the first 18 parts of every road image. All parts



(a) (b) (c) (d)

Figure 6. First pair of pictures: User clicks per screen required to obtain correct road mapping subject to number of image parts already seen by our learnable detection algorithm (a) - *road1*, *road2* data (b) - *road3* and *road4* data. Solid squares show necessary number of clicks if our detection algorithm is applied before user input. Empty squares show necessary number of clicks without use of automatic detection algorithm. Y-axis shows estimated number of clicks and x-axis represents number of processed images. Second pair of pictures: Error of automatic detection algorithm subject to number of image parts already seen by learnable detection algorithm (c) - *road1*, *road* data (d) - *road3* and *road4* data. The error is measured as a fraction of image square misclassified by our detection algorithm. At all plots green lines correspond to *road1* and blue lines correspond to *road2*. Y-axis shows error rate and x-axis represents number of processed images.

are about 0.5 megapixel size and correspond to 5-10 meters of road surface. Some of image parts with results of our automatic detection are shown in Figure 8.

Experiments setup

We have developed a testing framework which emulates user activity at the on-line stage. Given the classification results and ground truth data it starts with automatic thresholds adjusting. Gradient descent algorithm is used to determine a set of thresholds that minimizes total area of misclassified objects. Then user interaction is emulated as follows. At first, our framework corrects all errors of automatic detection which can be amended by relabeling segments of the coarsest segmentation scale. Then testing framework emulates user-aided error correction at subsequent segmentation scale. This procedure is repeated up to the most detailed segmentation scale. Total number of clicks required for errors correction is calculated as a sum of click counts at all segmentation scales. This statistic measures overall usability of our tool for road mapping.

One can see total clicks count per image part measured on 4 roads from our image base in Figure 6 (a, b). *Road1* and *road2* contain greater number of road defects than *road3* and *road4*, therefore larger number of user clicks is required for mapping last two roads. We have compared number of clicks required to achieve accurate mapping when user corrects errors of our automatic detection algorithm with the number of clicks required for mapping road surface from scratch when no automatic detection is performed. One (?- или It can be seen) can see that usage of automatic detection algorithm leads to advance in usability of mapping tool.

As a matter of fact, road marking can be usually found perfectly after processing the second or the third part image. So, the problem of road defects detection is more challenging. Figure 6 (c, d) demonstrates misclassified area of road defects subject to number of image parts seen by detection algorithm.

Figure 7 illustrates false positive and false negative error rates of road defect by pixels on *road1* data. This picture represents usual behaviour of our system. The rate of detected defects increases over time when more defects examples shown to automatic detection algorithm.

In summary, overall error tends to decrease while the number of handled images grows. The system usually starts to distinguish road defects since two or three images have been handled. Some road images like *road3* and *road4* contain a small amount of road defects (some image parts do not contain them at all). Although learning process is slowed down and benefit of using interactive system is reduced on such kind of roads however, usage of automatic detection result still remains beneficial.

8. CONCLUSIONS AND FUTURE WORK

We have presented a tool for efficient interactive mapping of road defects and lane marking on rectified images of road pavement surface. Intensive use of computer vision methods on different stages of our data processing workflow increases usability of the tool.

The most significant drawbacks of our tool is the limitation of using segments in user interaction stage and incapability to correct detection results on sub-segment level. Also our system currently is unable to accommodate to changes of the road structure, e.g. illumination level changing. This drawback can be eliminated if we provide on-line classifier with concept adapting.

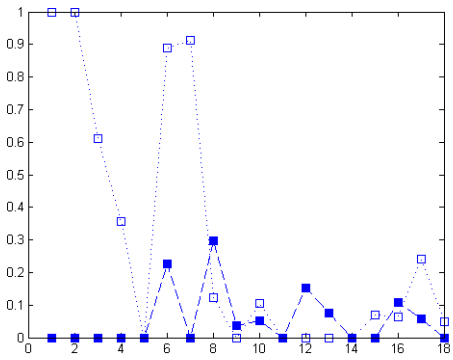


Figure 7. False positive and false negative rates on *road1* data subject to a number of handled road sections. Y-axis shows error rate and x-axis represents number of processed images.

9. REFERENCES

Breiman, L., 2001. Random Forests. *Machine Learning*, 45(1), pp. 5–32.

Criminisi, A., Shotton, J., Winn, J., Rother, C., 2006. TextonBoost: Joint Appearance, Shape and Context Modeling for Multi-Class Object Recognition and Segmentation. *In proc. of European Conference on Computer Vision,* Graz, Austria.

Domingos, P., Hulten, G., 2000. Mining high-speed data streams. *In proc. of the VI ACM SIGKDD International Conference on Knowledge Discovery and Data Mining,* Boston, USA. pp. 71 - 80.

Land, E.H., McCann, J.J. 1971. Lightness and Retinex Theory. *Journal of the Optical Society of America*, 61(1), pp. 1-20.

Leung, T, Malik, J. 1999. Recognizing Surfaces Using Three-Dimensional Textons. *In proc. of the International Conference on Computer Vision*, Kerkyra, Corfu, Greece Vol. 2, pp. 1010-1118,

Oza, N.C. 2005. Online Bagging and Boosting. *In proc. of IEEE International Conference on Systems, Man, and Cybernetics, Special Session on Ensemble Methods for Extreme Environments.* - New Jersey, USA, vol. 3, pp. 2340–2345

Paris, S., Durand, F. 2007. A Topological Approach to Hierarchical Segmentation using Mean Shift. *In proc. of Computer Vision and Pattern recognition*, Minneapolis, Minnesota, USA.

Sudakov, S., Barinova, O., 2008, Velizhev, A., Konushin, A. Semantic segmentation of road images based on cascade classifiers, *In proc. of Pattern Recognition an Image Analysis*, Nizhny Novgorod, Russia. vol. 2, pp. 112-116.

Viola, P., Jones, M. 2002, Robust Real-time Object Detection, *International Journal of Computer Vision*.

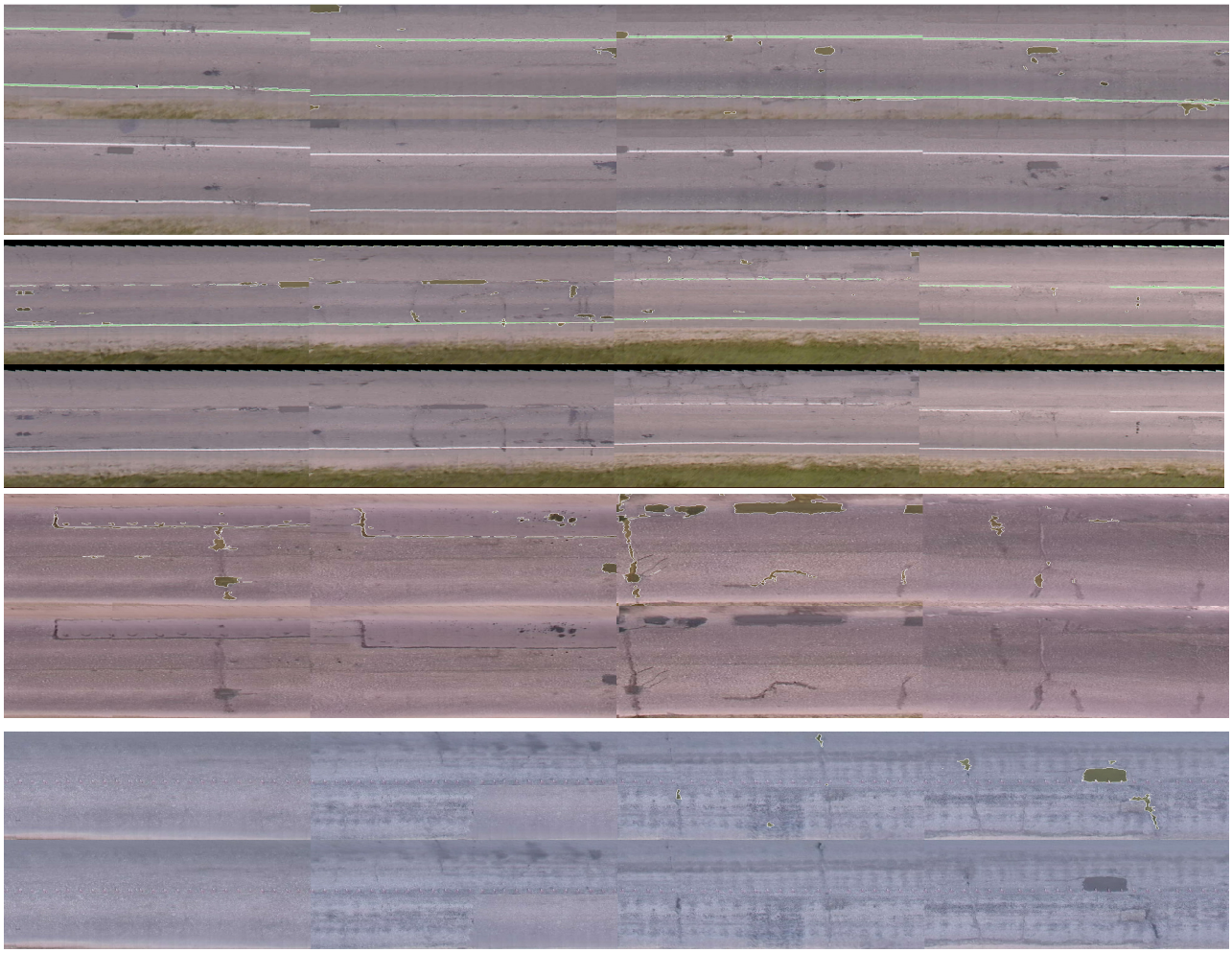


Figure 8. Results of automatic detection of road defects and lane marking. From top to bottom: *road1* data, *road2* data, *road3* data, *road4* data. Image parts number 3, 5, 10 and 18 are shown together with automatic detection results before manual correction. Lane marking is shown in green with blending, road defects are shown in brown with blending. Picture is better viewed in color and magnified.

SURFACE MODELLING FOR ROAD NETWORKS USING MULTI-SOURCE GEODATA

Chao-Yuan Lo*, Liang-Chien Chen, Chieh-Tsung Chen, and Jia-Xun Chen

Department of Civil Engineering, National Central University, Jhungli, Taoyuan 32001, Taiwan – freezer@csrsr.ncu.edu.tw
Center for Space and Remote Sensing Research, National Central University, Jhungli, Taoyuan 32001, Taiwan – lcchen@csrsr.ncu.edu.tw
Department of Land Administration, M.O.I., Taipei 10055, Taiwan – {moi5383; moi1240}@moi.gov.tw

Commission III, WG III/4

KEY WORDS: Surface, Reconstruction, Three-dimensional, Geometric, Laser scanning, Modelling

ABSTRACT:

Road systems are the fundamental component in the geographic information systems. This kind of civil infrastructures has large coverage and complex geometry. Thus, the modelling process leads to handling huge data volume and multi-source datasets. A reasonable process should be able to reconstruct separate parts of road networks and combine the surfaces together. Hence, the reconstruction of complete three-dimensional road networks needs scrutiny when a large area is to be processed. This paper proposes a scheme to focus on this issue using an integrated strategy with multi-source datasets. The modelling processes combine different data sources to refine road surfaces to keep the continuities in elevation and slope. The proposed scheme contains three parts: (1) data pre-processes, (2) planimetric networking, and (3) surface modelling. In the first part, datasets are registered in the same coordinate system. In the next step, topographic maps provide the roadsides to derive the geometric topology of road networks. Finally, those centerlines combine airborne laser scanning data to derive road surfaces. Considering the data variety, some road segments generated from aerial images are also included in the proposed scheme. Then, the successive process integrates those models for the refinement of road surfaces. The test area is located in Taipei city of Taiwan. The road systems contain local streets, arterial streets, expressways, and mass rapid transits. Some roadways are multi-layer and cross over with different heights. The final results use three-dimensional polylines and ribbons to represent geometric directions and road surfaces. Experimental results indicate that the proposed scheme may reach high fidelity.

1. INTRODUCTION

Based on the viewpoint of decision support for modern cities, the reconstruction of a virtual environment is an essential task. The applications include urban planning, traffic simulation, true orthorectification (Zhou et al., 2005), hazard simulation, communication, etc. Since the road models are one of the most prominent components in the urban information systems, the reconstruction of the model becomes increasingly important. In general, the traditional topographic map is a kind of widely used dataset that describes road geometries. It can efficiently build single-layer road models. However, this civil infrastructure is developed rapidly in modern cities for the traffic demand, and road types become more complex including local streets, arterial streets, expressways, freeways, and mass rapid transit. Single-layer road networks have changed to multi-layer systems and topomaps may be insufficient to describe complex roads. The elevation information of road surfaces needs to be considered for the separation of overpasses.

Some researches focused on the surface modelling processes with different strategies and data, e.g. aerial photos, laser-scanning data, GPS data, topomaps, and so on. Cannon (1992) proposed a scheme to locate the three-dimensional road profiles integrating GPS and INS data. A related work also had been made to estimate the slope information of road profiles using GPS data (Han and Rizos, 1999). Some studies preferred to derive road information in spectral domain. They analyzed road shapes of centerlines or boundaries to derive road geometries with vehicle-based images (Yan et cl., 2008), aerial photos (Treash and Amaratunga, 2000; Hinz and Baumgartner, 2003; Dal Poz et al., 2004), satellite images (Yan and Zhao, 2003; Doucette et al., 2004; Hu et al., 2004a; Kim et al., 2004; Karimi and Liu, 2004; Yang and Wang, 2007), airborne laser scanning data (Clode et al., 2007). Some proposed semi-automatic approaches basing on the matching technique to reliably extract road geometries with manual editing from high-resolution satellite imagery (Hu et al., 2004a; Kim et al.,2004). Easa et al. (2007) focused on the automatic image processing to extract edge lines for calculation of geometric parameters to describe horizontal alignments from high resolution images.

On the other hand, an integrating strategy had been proposed to deal with this issue using aerial images and laser scanning data (Hu et al., 2004b; Zhu et al., 2004). Zhang (2003) integrated aerial photos and geo-database to derive and update three-dimensional road data. Moreover, geo-database and laser scanning data also could be a combination. Hatger and Brenner (2003) calculated the profile geometries of centerlines from the geo-database and digital surface models. The segment-based method used region growing to detect road areas for the calculation of geometric parameters to refine the geo-database. Furthermore, Cai and Rasdorf (2008) also combined two datasets, airborne laser scanning data and planimetric centerline

* Corresponding author

data, to establish three-dimensional centerlines. Those elevation differences of multi-layer areas were marked with an additional attribute.

Other researches based on the mapping concepts to regard road surfaces as some parts of terrain. Thus, some filtering techniques were developed to extract ground information from airborne laser scanning data for DEM (digital elevation model) generation. The performances of those filtering methods had been compared by Sithole and Vosselman (2004). In some cases, single-layer road network could be regarded as a part of bare earth. Hu (2003) assumed that road profiles could be piecewise continuous and extract road points with the elevation threshold from discrete point clouds. Vosselman (2003) used laser scanning data to reconstruct single-layer road models referencing cadastral maps. This process derived road points within road areas first and generated models with triangular irregular network (TIN) surface. The refinement step assumed that road surfaces without slope, curvature, or torsion and smoothed them with the second order constrained polynomial functions. Additionally, Sithole and Vosselman (2006) handled the multi-layer condition which point clouds of overpasses were marked with the analyses of slope and elevation difference. They regarded those marked areas as the extended parts of terrain so that there was at least one side should connect to the ground. Oude Elberink and Vosselman (2006) paid attention to multi-layer interchanges using laser scanning data and topographic maps. Those roads were TIN-based models, and the multi-layer parts were separated into different elevations. Chen and Lo (2009) proposed a scheme to fuse airborne laser scanning data and topographic maps. The planimetric geometry and elevation of each road segment were established. The road models were represented as vector-based ribbons.

As a summary, the integration of heterogeneous datasets seems to be a popular way to reconstruct three-dimensional road models, especially topomaps and laser scanning data. Most studies focused on the modelling processes for single-layer road systems, and few of them discussed about multi-layer parts. The reconstruction of road systems using a robust method for the large coverage is still an ongoing topic. Although the proposed scheme (Chen and Lo, 2009) reconstructed multi-layer models, this sequential modelling process was a local approach to smooth the model surfaces. In a rigorous way, we may need to consider a method to handle complete road networks and preserve the capacity for model updating.

This investigation proposes an approach to model three-dimensional road networks using laser scanning data and topographic maps. Because some countries may have complete information of road boundaries and centerlines, others may use CAD data to describe roads using piecewise polylines in planimetric domain without geometric topology. Therefore, this investigation needs to compute topology of road networks and derive road elevations from discrete point clouds. In this planimetric part, each road segment would be generated its centerline and connect to others for network topology with conjunction points. The successive processes then include laser scanning data to derive road surfaces of each segment and refine all conjunction points to maintain the continuities in elevation and slope. When road systems encounter changes over time, new roads for example, they are needed to rebuild according to the latest dataset. Those new parts are digitized from aerial photos in this modelling process and refined their elevations with existed models to keep the system coincidence. The results are to be represented as three-dimensional ribbons.

2. METHODOLOGY

Based on the viewpoint of surface modelling, we integrate multi-source datasets to reconstruct complete surface modelling. In this investigation, we assume that the vertical and horizontal alignments of each road segment are continuous within a local area. Moreover, a global approach implements B-spline surface fitting refines the elevations of network conjunctions by keeping the continuities. The local approach sequentially modifies the elevation of each road segment. The proposed scheme has also considered the multi-layer condition. The processes have three parts: (1) registration, (2) planimetric networking, (3) model surfacing. The first part is to register all datasets, i.e. topomaps, laser scanning data, and three-dimensional boundaries. The next step then produces the networks using roadsides from topographic maps. The third part computes the model surface of each road segment and combines all roads from different sources to refine their vertical and horizontal profiles to keep the continuities in elevations and slope. The workflow shows in Figure 1.

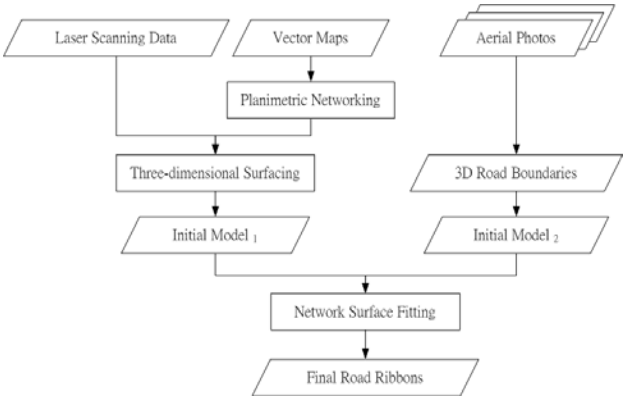


Figure 1. Workflow

2.1 Planimetric networking

In traditional CAD-based topographic maps, there are several road levels like local streets, arterial streets, expressways, etc. This kind of data records those boundaries using piecewise polylines. In addition, the topomaps may lack some information, e.g. attributes, topology, and centerlines. To directly use the topomaps for centerline generation is still difficult if those boundaries are independent without pair relationship. Therefore, this step uses those existed boundaries to compute centerlines for the reconstruction of network topology.

First of this part, the planimetric process separates those boundaries into many simple straight lines. Those pieces then connect to each other according to the empirical thresholds of distance and angle for the development of complete boundary lines. The second step pairs those produced edges to position centerlines. All the planar conjunctions, i.e. crossroads, are automatically added a node point to split those centerlines and establish the topology, besides overpasses. The networking procedure would detect those multi-layer parts with boundary analysis (Chen and Lo, 2009) and mark which centerlines go through those areas.

2.2 Three-dimensional surfacing

After planimetric networking, laser scanning data is employed for road surfacing. The airborne LIDAR data records plenty discrete points with accurate elevation information. This surfacing step, basing on the planimetric geometry, extracts

those points on the road surfaces and computes their elevations. Chen and Lo (2009) assumed that the local relief of road surfaces should be continuous for traffic. They then analyzed the elevation histogram and extract the candidate points on the roads to fit surfaces. The proposed scheme contains two parts: (1) initial modelling and (2) profile refinement. The initial process calculates the surface elevations from discrete points along each produced centerline. Those original road surfaces then are modified their elevations to keep the model continuities in elevation and slope.

2.3 Initial Modelling

Those existed airborne laser scanning data describe accurate elevations with considerable quantities of points. The laser beam also has the opportunity to penetrate canopies to detect elevations in occluded areas. However, this kind of data has no distinct boundaries. To directly use discrete points for surface modelling is a difficult work. Chen and Lo (2009) proposed a two-way method to extract road points. They assume that the road surface profile is smooth and continuous in a local area so that the maximum number of elevation histogram of road points may locate within a certain interval. One process, thus, extracts points with a designed threshold to fit surfaces at each vertex of centerlines. The used equations, i.e. linear and quadratic polynomial functions, are shown in Equation (1) and (2). The unknown parameters are the s11~ s26. Those two hypotheses are automatic selected according to the analysis of the standard deviation during the fitting procedure. In some conditions, road surfaces may be interfered by cars or canopies, this threshold may lead to remove too many points to calculate surfaces. The other process then selects the locally lowest point to be the surface elevation. In the first way, the cross-section is a curve to represent the reality of horizontal profiles by surface fitting. On the other hand, the second way provides a flat road surfaces.

$$S_1(Z) = s_{11} + s_{12}X + s_{13}Y \tag{1}$$

$$S_2(Z) = s_{21} + s_{22}X + s_{23}Y + s_{24}XY + s_{25}X^2 + s_{26}Y^2 \tag{2}$$

where X, Y,and Z are coordinates of the LIDAR points; and S_{11}~S_{26} are parameters of the surface function.

2.4 Profile refinement

This investigation describes each road segment with two nodes and several vertices, i.e. conjunction points of networks and consecutive center points, respectively. In the previous step, we independently derive the elevation of each vertex from original point clouds. Nevertheless, some parts of each vertical profile may be discontinuous, erroneous, and empty. The following process then transforms the coordinates (X, Y, Z) to mileages (Stations) and fine-tunes the vertical profiles with three mathematical models. The linear, quadratic, or cubic functions are used to refine its vertical profile. The mathematical models are formulated in Equation (3), (4), and (5), respectively. The modification process would select an optimal function according to the minimum standard deviation. Those errors and empty values of each road segment are detected and re-computed.

After vertical refinements, the continuities of horizontal profiles may be interfered. In this process, the surface fitting then includes those consecutive vertices to smooth their elevations. Equation (1) and (2) are considered in the smoothing process. However, if a road segment is too long, the used models may be insufficient to describe the characteristics of vertical profiles. Chen and Lo (2009) considered that road systems are designed and organized by low-ordered polynomial models everywhere so that the theoretical models can easily represent each sub part of one vertical profile. They created some pseudo nodes for each road segment and smooth the geometry of cross-sections with Equation (1) or (2). The profile refinement is an iterative process until the elevation change of each road segment is smaller than the designed tolerance.

$$L_1(H) = p_{11} + p_{12}M \tag{3}$$

$$L_2(H) = p_{21} + p_{22}M + p_{23}M^2 \tag{4}$$

$$L_3(H) = p_{31} + p_{32}M + p_{33}M^2 + p_{34}M^3 \tag{5}$$

where p_{11}~p_{34} are parameters of the line function; M is the mileage of each road segment; and H is vertex height.

2.5 Network surface fitting

This study focuses on the modelling procedure with multiple roads using different data and keeps the results continuous in elevation and slope. For this purpose, we propose to use B-spline surface fitting to modify all conjunction points of road networks. The elevation correction of each road segment is then re-arranged to its internal vertices. In this step, we simplify the format of those produced road models for surface fitting. The conjunction points are selected and computed their new elevations using B-spline curve function, i.e. Equation (6), to maintain the continuities in elevation and slope. After the fitting process, all the conjunctions have new height values, and elevation changes then bring to each road segment and modify the elevations of internal vertices. The iteration stops when the elevation change of all road systems is smaller than the threshold. In short, the proposed scheme makes the capability to reconstruct three-dimensional road models combining different data sources.

$$C(u) = \sum_{i=0}^{n} f_i(u) P_i \tag{6}$$

where P_j are control points and f_j is a basis function.

3. EXPERIMENTAL RESULTS

The scheme was validated using data for single and multiple layer road systems in Taipei City of northern Taiwan. The area has the coverage of 3,200m*6,600m. The test site includes arterial streets, local streets, expressways, and mass rapid transit in an urban area. The test datasets include topographic maps, airborne scanning data, and three-dimensional boundaries. The scale of the topographic maps is 1:1000. They contain several feature layers, such as buildings, roads, power lines, etc. In Taiwan, road boundaries are recorded as independent planimetric polylines without topology or transportation attributes, as shown in Figure 2. As shown in Figure 3, the LIDAR data was derived from a Leica ALS50 system in March 2007. The flight altitude ranged from 1200 to 1500 m. The laser pulse rate was 70 kHz, and the point density was about 10 points/m^2. The random error of laser points in elevation is better than 0.15m (ITRI, 2006). The third type dataset is the three-dimensional road boundaries which were digitized from aerial images. The spatial resolution of used DMC images is about 17 cm. Figure 4 shows edited road boundaries in aerial images.

LIDAR road points are incorporated to provide height information for three-dimensional surface modelling. A threshold for the maximum elevation histogram is used to determine which method will be used for road surface initialization, i.e., either surface fitting or lowest point selection. The radius of a buffer circle is set to be half the roadwidth. Based on the experience, this percentage threshold of elevation histogram of initial surface modelling would be 30%. The reason is that the space interval of the along-track vertices is densified to 0.5m. Those local slopes of the vertical profiles are assumed to be less than 45°, i.e., the elevation change is smaller than 1.5m. Since the slope of a road is seldom larger than 45°, the threshold is reasonable that adapts for general applications. Possible interference could be the presence of dense vehicles that make the point clouds deviate from the road surface. This could lead to unreliable results. For more precise surface modelling, the spanning distance between pseudo nodes is set to be 200 m, according to the rules of roadway designs. Next, the vertical profiles, cross-sections, and intersections are smoothed according to either height difference or iterative times. Figure 5 shows the reconstructed three-dimensional road models.

To evaluate the reconstructed road models, reference LIDAR road points are extracted manually. The normal height differences between the reference points and the reconstructed surfaces are compared to calculate the relative error assessment. The index of modelling error is expressed as the root mean square error (RMSE). The generated results indicate that the RMSEs for the modelled surfaces of test sites are lower than 0.15 m. Those values indicate that the iteratively local approach may lead to modelling errors within the range of random error of the raw data. The slopes of reconstructed models in vertical profiles and cross-sections are estimated and shown in Figure 6 and 7, respectively.

Figure 2. Road boundaries in topographic maps

Figure 3. Sub part of laser scanning data

Figure 4. Digitized road boundaries in aerial images

Figure 5. One part of reconstructed models (Overpass)

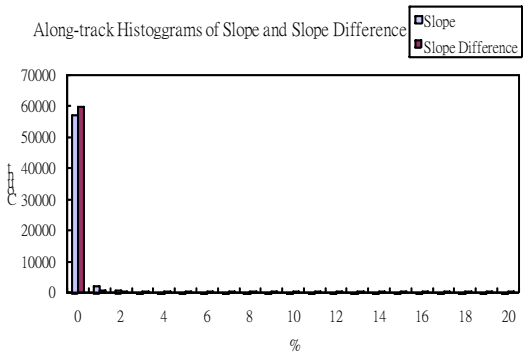


Figure 6. Histogram of slope and slope difference of along-track profiles

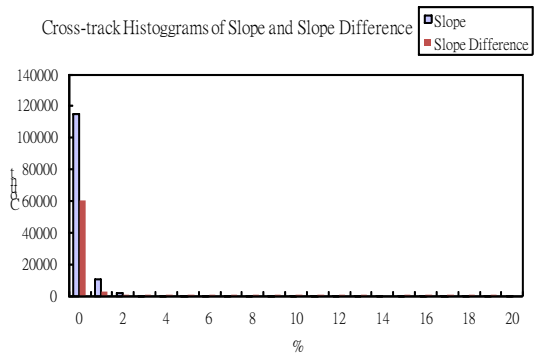


Figure 7. Histogram of slope and slope difference of cross-track profiles

4. CONCLUSIONS

Without considering the landuse changes, the proposed scheme integrates topographic maps, airborne laser scanning data, and three-dimensional road boundaries for 3D surface modelling. In the modelling process, road surfaces are initialized from the raw LIDAR data. Additionally, we proposed a network surface fitting process to refine those model surfaces from multi-source datasets to maintain the continuities in elevation and slope. The test site includes local streets, arterial streets, and expressways to validate the ability of the proposed scheme. According to the experimental results, the three-dimensional surface modelling accuracy reaches 0.149 m. In addition, the modelling results indicate that this approach reaches an error, which is within the random error of the raw data.

5. ACKNOWLEDGEMENT

This research is partially funded by Dept of Land Administration, M. O. I. of Taiwan. The authors would also thank the same organization for providing airborne laser scanning data, topographic maps, and aerial photos.

6. REFERENCES

Cannon, M. E., 1992. Integrated GPS-INS for High-accuracy Road Positioning, *Journal of Surveying Engineering*, Vol. 118, No. 4, pp. 103-117.

Cai, H., and Rasdorf, W., 2008. Modelling Road Centerlines and Predicting Lengths in 3-D Using LIDAR Point Cloud and Planimetric Road Centerline Data, *Computer-Aided Civil and Infrastructure Engineering*, Vol. 23, pp.157–173.

Chen, L.C., and Lo, C.Y., 2009. three-dimensional Surface modelling Via The Integration Of Large-Scale Topomaps And Airborne Lidar Data, *Journal of the Chinese Institute of Engineers*, Vol. 32, No. 6, unpaginated.

Clode, S., Rottensteiner, F., Kootsookos, P., and Zelniker, E., 2007. Detection and Vectorization of Roads from LIDAR Data, *Photogrammetric Engineering and Remote Sensing*, Vol. 73, No. 5, pp. 517-535.

Dal Poz, A. P., Vale, G. M., and Zanin, R. B., 2004. Automated Road Segment Extraction by Grouping Road Objects, Proceedings of the XXth ISPRS Congress, Istanbul, Turkey, Vol. XXXV(B3), pp. 436-439.

Doucette, P., Agouris, P., and Stefanidis, A., 2004. Automated Road Extraction from High Resolution Multispectral Imagery, *Photogrammetric Engineering and Remote Sensing*, Vol. 70, No. 12, pp. 1405-1416.

Easa, S. M., Dong, H. B., and Li, J., 2007. Use of Satellite Imagery for Establishing Road Horizontal Alignments, *Journal of Surveying Engineering*, Vol. 133, No. 1, pp. 29-35.

Han, S., and Rizos, C., 1999. Road Slope Information from GPS-derived Trajectory Data, *Journal of Surveying Engineering*, Vol. 125, No. 2, pp. 59-68.

Hatger, C., and Brenner, C., 2003. Extraction of Road Geometry Parameters from Laser Scanning and Existing Databases, Proceedings of the ISPRS Workshop Laser scanning 2003, Dresden, Germany, unpaginated CD-ROM.

Hinz, S., and Baumgartner, A., 2003. Automatic Extraction of Urban Road Networks from Multi-view Aerial Imagery, *ISPRS journal of Photogrammetry and Remote Sensing*, Vol. 58, pp. 83–98.

Hu, X., Zhang, Z., and Tao, C. V., 2004a. A Robust Method for Semi-automatic Extraction of Road Centerlines Using a Piecewise Parabolic Model and Least Square Template Matching, *Photogrammetric Engineering and Remote Sensing*, Vo. 70, No. 12, pp. 1393-1398.

Hu, X., Tao, C. V., and Y. Hu, 2004b. Automatic Road Extraction from Dense Urban Area by Integrated Processing of High Resolution Imagery and LIDAR Data, Proceedings of the ISPRS XXth Congress, Istanbul, Turkey, ISPRS Workshop Commission III, unpaginated CD-ROM.

Hu, Y., 2003. Automated Extraction of Digital Terrain Models, Roads and Buildings Using Airborne LIDAR Data, PhD dissertation, Department of Geometrics Engineering, University of Calgary, 206 p., URL: http://www.geomatics.ucalgary.ca/links/GradTheses.html (last date accessed: 30 May 2005)

ITRI, 2006. High Accuracy and High Resolution DEM Mapping and Database Establishment for Selected LIDAR Survey Areas and the Developments of Their Applications, Technical report, *Energy & Resources Laboratories of Industrial Technology Research Institute*, 316 p. (in Chinese).

Karimi, H. A., and Liu, S., 2004. Developing an Automated Procedure for Extraction of Road Data from High-Resolution Satellite Images for Geospatial Information Systems, *Journal of Transportation Engineering*, Vol. 130, No. 5, pp. 621-631.

Kim, T., Park, S. R., Kim, M. G., S. Jeong, and Kim, K. O., 2004. Tracking Road Centerlines from High Resolution Remote Sensing Images by Least Squares Correlation Matching, *Photogrammetric Engineering and Remote Sensing*, Vol. 70, No. 12, pp. 1417-1422.

Oude Elberink, S., and Vosselman, G., 2006. three-dimensional Modelling of Topographic Objects by Fusing 2D Maps and Lidar Data, International Archives of Photogrammetry, *Remote Sensing and Spatial Information Sciences*, Vol. 36, part 4, Goa, India, 6p. (on CD-ROM).

Sithole, G., and Vosselman, G., 2004. Experimental Comparison of Filter Algorithms for Bare-earth Extraction from Airborne Laser Scanning Point Clouds, *ISPRS Journal of Photogrammetry and Remote Sensing*, Vol. 59, pp. 85–101.

Sithole, G., and Vosselman, G., 2006. Bridge Detection in Airborne Laser Scanner Data, *ISPRS Journal of Photogrammetry and Remote Sensing*, Vol. 61, No. 1, pp. 33-46.

Treash, K., and Amaratunga, K., 2000. Automatic Road Detection in Grayscale Aerial Images, *Journal of Computing in Civil Engineering*, Vol. 14, No. 1, pp. 60-69.

Vosselman, G., 2003. three-dimensional Reconstruction of Roads and Trees for City Modelling, International Archives of Photogrammetry, *Remote Sensing and Spatial Information Sciences*, Vol. XXXIV-3/W13, pp. 231-236.

Yan, W., Li, B., and Fairhurst, M., 2008. Robust Surface modelling and Tracking Using Condensation, *IEEE Transactions On Intelligent Transportation Systems*, Vol. 9, No. 4, pp. 570-579.

Yan, D., and Zhao, Z., 2003. Road Detection from QuickBird Fused Image Using IHS Transform and Morphology, *International Geoscience and Remote Sensing Symposium*, Vol. 6, pp. 3967-3969.

Yang, J., and Wang, R. S., 2007. Classified Road Detection from Satellite Images Based on Perceptual Organization, *International Journal of Remote Sensing*, Vol. 28, No. 20, pp. 4653-4669.

Zhang, C., 2003. Towards an Operational System for Automated Updating of Road Databases by Integration of Imagery and Geodata, *ISPRS Journal of Photogrammetry and Remote Sensing*, Vol. 58, No. 3-4, pp. 166-186.

Zhou, G., Chen, W. R., Kelmelis, J. A., and Zhang D. Y., 2005. A Comprehensive Study on Urban True Orthorectification, *IEEE Transactions on Geoscience and Remote Sensing*, Vol. 43, No. 9, pp. 2138-2147.

Zhu, P., Lu, Z., Chen, X., Honda, K., and Eiumnoh, A., 2004. Extraction of City Roads Through Shadow Path Reconstruction Using Laser Data, *Photogrammetric Engineering and Remote Sensing*, Vol. 70, No. 12, pp. 1433-1440.

AUTOMATIC EXTRACTION OF URBAN OBJECTS FROM MULTI-SOURCE AERIAL DATA

Adriano Mancini, Emanuele Frontoni and Primo Zingaretti

Dipartimento di Ingegneria Informatica Gestionale e dell'Automazione
Università Politecnica delle Marche
Ancona, ITALY
{mancini,frontoni,zinga}@diiga.univpm.it

KEY WORDS: LiDAR, buildings, road extraction, automated classification, city models

ABSTRACT:

Today, one of the main applications of multi-source aerial data is the city modelling. The capability to automatically detect objects of interest starting from LiDAR and multi-spectral data is a complex and an open problem. The information obtained can be also used for city planning, change detection, road graph update, land cover/use. In this paper we present an automatic approach to object extraction in urban area; the proposed approach is based on different sequential stages. The first stage basically solves a multi-class supervised pixel based classification problem (building, grass, land and tree) using a boosting algorithm; after classification, the next step provides to extract and filter land areas from classified data; the last step extracts roundabouts by the Hough transform and linear roads by a novel approach, which is robust to noise (sparse pixels); the final representation of extracted roads is a graph where each node represents a cross between two or more roads. Results on a real dataset of Mannheim area (Germany) using both LiDAR (first - last pulses) and multi-spectral high resolution data (Red - Green - Blue - Near Infrared) are presented.

1 INTRODUCTION

TODAY the availability of high spatial resolution LiDAR and multi-spectral data collected by aerial vehicles (manned or unmanned) traces new ways for the possible applications. City modeling, object extraction (e.g., buildings, roads, bridges, ...), urban growth analysis, land use/cover, developing 3D models, are the main studied applications. Usually the analysis of data is made by a human operator; traditional photo-interpretation is a slow and expensive process that requires specialized experts; accuracies similar to those of man-made maps can now be reached by automatic object extraction and classification approaches, but with considerably less wasted time and money, thus allowing high update rates.

The ability to automatically classify data starting from a set of heterogeneous features is fundamental to design an automatic approach. One of the first method used to classify LiDAR data was the height threshold to a normalized DSM (nDSM) (Weidner and Forstner, 1995); using this method it is possible to extract objects as buildings, but its has a lot of well-known drawbacks: high-density canopy can be classified as building and it is not possible to distinguish low height objects as lands or roads. Multi-spectral data allow to extend the set of classified objects producing higher accuracy. Many machine learning approaches were adopted to solve the problem of object extraction from multi-source data; Bayesian maximum likelihood method (Walter, 2004), Dempster-Shafer (Lu et al., 2006), boosting using AdaBoost (Frontoni et al., 2008).

Common objects as buildings or roads are the main interesting features that can be extracted from the classified data; road extraction is a classical problem of remote sensing, but not completely solved. A really interesting overview (updated to 2003) can be found here (Mena, 2003). Using only multi-spectral data (Bacher and Mayer, 2005), road extraction is an extremely difficult task especially in urban area also using high-resolution imagery as IKONOS or SPOT. Problems as occlusion (due to the presence of trees), noise inducted by vehicles or object shadows, influence the quality of road extraction; moreover, spectral separability of road respects to other objects (e.g. bituminous roofs) is not always guaranteed. Snakes/active contours are classical methodological tools; different version of standard snake (Kass et al., 1987) were developed to solve the problem of road extraction especially in not urban area (Marikhu et al., 2006). Moreover this approach requires a wide set of good seed points, which are often user defined. The fusion of LiDAR and multi-spectral data is a powerful tool for road extraction; LiDAR helps to distinguish between high objects as buildings or canopies, while multi-spectral data allow to distinguish between land/road and grass or other low profile objects (Clode et al., 2005). SAR imagery can be also useful for road extraction with results comparable with LiDAR (Guo et al., 2007). However the goodness of LiDAR and multi-spectral data fusion approaches allows to obtain interesting results in building / road extraction.

In this paper, a classification approach, using boosting classifier to fuse LiDAR and multi-spectral data, is presented. The AdaBoost technique with CART classifier as weak learner, classifies data distinguishing among four classes: building, grass, land and tree; the ReliefF (Liu and Motoda, 2008) feature selection algorithm allows to consider only meaningful features to minimize the misclassification. The result of classification stage is then used to extract buildings, roads and roundabouts; the approach here proposed extracts and clusters a set of linear roads using a pyramidal representation to reduce time and memory usage. The procedure is totally automatic and requires only a minimum interaction with user; a user-defined training set is necessary to train the classifier and control the learning accuracy; the training set often can be directly accessible by a web-GIS or a photo-interpretation process over a very small portion of global area; we use a training set that covers less than 0.5% of total area.

The paper is organized as follows. Section 2 introduces the methodology for classification and object extraction; Section 3 explains the data set used for experiments, the adopted classifer and the classification results on a four class problem. Section 4 presents the method and obtained results in road extraction; in Section 5 conclusions and future work are outlined.

2 METHODOLOGY

Building and road extraction, as mentioned above, require complex elaborations of multi-source data; we followed a multi-step procedure. The procedure here proposed consists of four sequential steps; the output of each module is the input for the following.

Step 1 - Feature generation. It calculates LiDAR and radiometric additional features for the classification stage; a total of seven mixed-features are currently adopted.

Step 2 - Classification. Using AdaBoost with a tree classifer as weak learner, it distinguishes among four main classes; a simple training set is adopted to train the classifer.

Step 3 - Object Extraction. It extracts buildings and/or roads from the classified data; in this paper we focus on road extraction and pre-filtering techniques;

Step 4 - Clustering. It is fundamental to model the extracted objects.

A graphical representation of discussed methodology is shown in Fig.1.

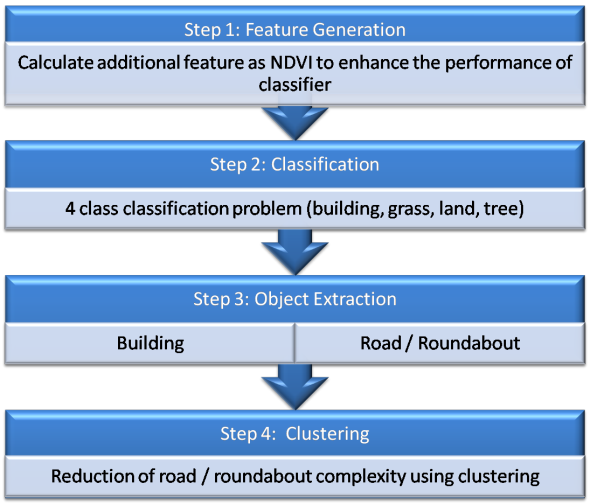


Figure 1: Methodology. The object extraction procedure has a hierarchical structure that simplifies the phase of result evaluation; different approaches can be easily tested without compromising the overall methodology

In the following sections, the results of each stage are presented; for completeness a deep results evaluation of building extraction is reported to evidence the quality of classification process; standard metrics are used to make in evidence the performance of AdaBoost classifier.

3 CLASSIFICATION

3.1 Dataset

The methodology presented in previous section, was validated in an urban area: LiDAR and multi-spectral data refer to the centre of the German city of Mannheim. This area is characterized with large buildings, mostly attached forming building blocks of different heights, many cars and little vegetation. Mannheim dataset has a resolution of 0.25m for the images and 0.5m for the range data; the total grid dimension is 1808 x 1452 (width x height).

The aerial images are orthorectified and four spectral bands are available: Red, Green, Blue, and Near InfraRed; laser range data consist of first and last pulse recordings acquired by an airborne laser scanner. Additional features were added to expand the feature space; main motivation is that using a feature weighting algorithm, is easy to find the best feature combination. Normalized Difference Vegetation Index (NDVI) and Green Normalized Difference Vegetation Index (GNDVI) were calculated. These indexes are useful to distinguish between some critical classes which LiDAR data cannot easily distinguish. Two pairs are critical: building/tree and land/grass. NDVI is a compact index which allows to better discriminate inside each cited pair. It is well known that canopies and grass have a NDVI value usually greater than 0.15, while for building and land classes is usually around or below zero. As introduced in the previous sections, we identified four main classes; for each class, we selected eight representative polygons. The total area of training set is below the 0.5%; it is useful to remark that the selection of these polygons is a low-time consuming activity that can be easily performed using a web-GIS or photo-interpretation (easy owing to the reduced number and kind of classes). The training set and a 3D view of the input dataset are shown in Figures 2 and 3.

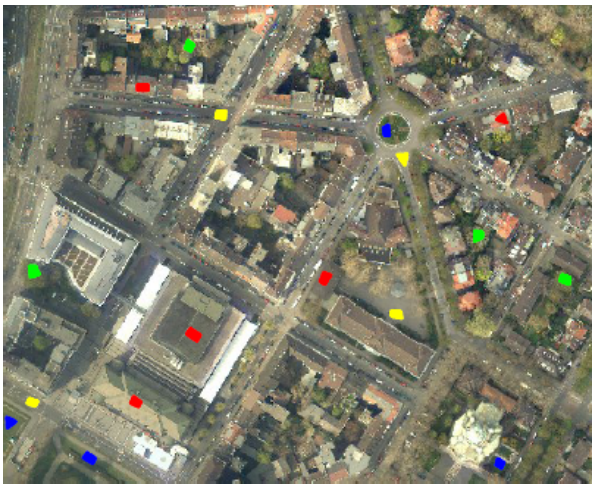


Figure 2: Data and Training set. Red stands for building, yellow for land, blue for grass and green for tree

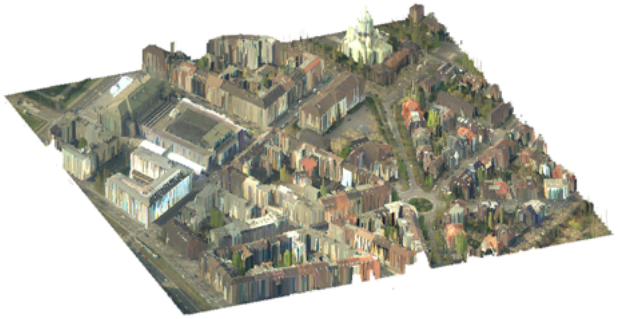


Figure 3: A 3D view of dataset; height of objects are obtained using the first pulse laser range data

The selected features used for classification are:

LiDAR: Δh is the height difference between the last pulse DSM and the DTM and Δp is the height difference between the first pulse and the last pulse DSM

Spectrals: R,G,B,NIR and NDVI (GNDVI is omitted because the weight associated to this feature was low)

The algorithm used for feature weighting was the ReliefF (Liu and Motoda, 2008); features with highest weights are Δh, Δp and NDVI; G B R and NIR have low weights; the goodness of selection is also demonstrated by the obtained results varying the set of features in the classification phase. The Weights obtained by the ReliefF algorithm are shown in Fig. 4.

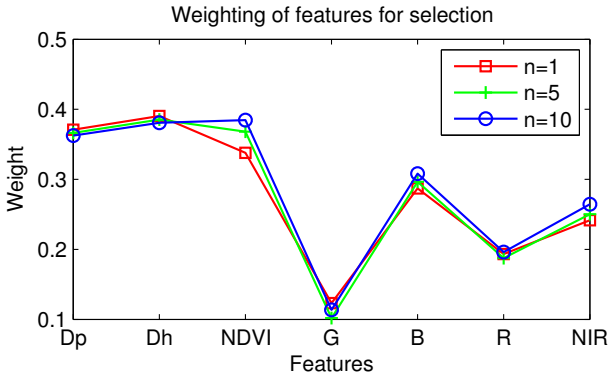


Figure 4: Results of ReliefF algorithm applied to the set of seven features; the n parameter represents the number of nearest instances from each class.

Analyzing the weight of each feature, it is evident as the LiDAR features Δp and Δh and the *NDVI* have the higher values; pure radiometric features do not allow to classify data correctly due to the lack of spectral separability.

3.2 Thresholding Normalized DSM

Thresholding Normalized DSM is a simple technique that allows to classify LiDAR data; only few objects can be extracted, mainly buildings. Problems which afflict this approach are the ambiguity of high density canopies and the impossibility to distinguish between land and grass. nDSM is defined as the subtraction of the DTM from the DSM of the same scene. A normalized DSM contains objects on a plane of height zero. Assuming that buildings in the scene have a known range of height, and that the heights of all other objects fall outside this range, buildings can be detected by applying appropriate height thresholds to the nDSM.

3.3 AdaBoost

AdaBoost (short for "adaptive boosting") is presently the most popular boosting algorithm. The key idea of boosting is to create an accurate strong classifier by combining a set of weak classifiers. A weak classifier is only required to be better than chance, and thus can be very simple and computationally inexpensive. Different variants of boosting, e.g. Discrete AdaBoost, Real AdaBoost (used in this paper), and Gentle AdaBoost (Schapire and Singer, 1999), are identical in terms of computational complexity, but differ in their learning algorithm. The Real AdaBoost algorithm works as follows: each labelled training pattern x receives a weight that determines its probability of being selected for a training set for an individual component classifier. Starting from an initial (usually uniform) distribution D_t of these weights, the algorithm repeatedly selects the weak classifier $h_t(x)$ that returns the minimum error according to a given error function. If a training pattern is accurately classified, then its chance of being used again in a subsequent component classifier is reduced; conversely, if the pattern is not accurately classified, then its chance of being used again is raised. In this way, the idea of the algorithm is to modify the distribution D_t by increasing the weights of the most difficult training examples in each iteration. The selected weak classifier is expected to have a small classification error on the training data. The final strong classifier H is a weighted majority vote of the best T (number of iterations) weak classifiers $h_t(x)$:

$$H(x) = sign\left(\sum_{t=1}^{T} \alpha_t h_t(x)\right)$$

It is important to notice that the complexity of the strong classifier depends only on the weak classifiers. The AdaBoost algorithm has been designed for binary classification problems. To deal with non-binary results we used a sequence of binary classifiers, where each element of such a sequence determines if an example belongs to one specific class. If the binary classifier returns a positive result, the example is assumed to be correctly classified; otherwise, it is recursively passed to the next element in this sequence; this techniques is known as "one against all". As weak classifer in this paper, a Classification And Regression Tree (CART) with three splits and $T = 35$ was used.

The CART method was proposed by (Breiman et al., 1984). CART produces binary decision trees distinguished by two branches for each decision node. CART recursively partitions the training data set into subsets with similar values for the target features. The CART algorithm grows the tree by conducting for each decision node, an exhaustive search of all available features and all possible splitting values; the optimal split is determined by applying a well defined criteria as Gini index or others ones (Duda et al., 2000).

3.4 Classification Results

In order to extract objects of interest from the previous described dataset, all the data were classified. In Fig. 5, the best result (in terms of detection rate) of classification using AdaBoost is shown. Moreover to evaluate correctly the quality of classification, a ground truth for buildings was manually created (see Fig. 6); the ground truth for the remaining classes actually is not available but it is planned to cover all the area to analyse exactly the classifier performance.

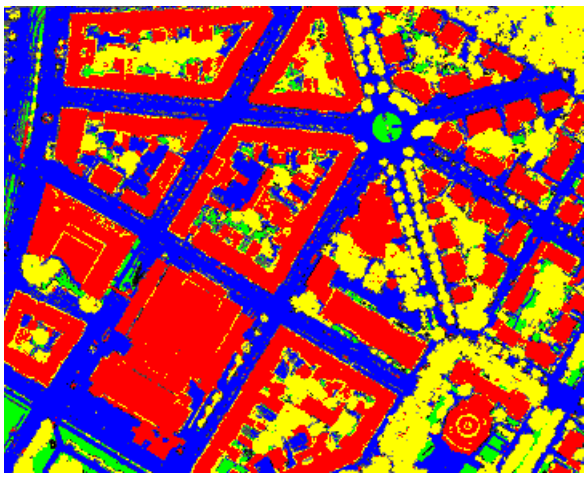


Figure 5: Results of classification using AdaBoost and the training set of 32 polygons; red stands for building, yellow for tree, blue for land and green for grass

In Table 1 the results for building extraction with different sets of features are highlighted; according to the weighting algorithm,

Figure 6: Ground truth used to evaluate the classification results; white pixels are buildings, blacks one are remaining objects

the combination of Δh, Δp and NDVI has the best performance in different indexes. A detailed description of indexes is[1]:

DR - Detection Rate: $DR = TP/(TP + FN + UP)$

FPR - False Positive Rate: $FPR = FP/(TN + FP + UN)$

FNR - False Negative Rate: $FNR = FN/(TP + FN + UP)$

UPR - Unclassified Positive Rate: $UPR = UP/(TP + FN + UP)$

OA - Overall accuracy: $OA = (TP + TN)/(TP + TN + FP + FN)$

R - Reliability: $R = TP/(TP + FP)$

TUR - Total Unclassified Rate: $TUR = (UP + UN)/(TP + TN + FP + FN + UP + UN)$

Classifier	DR	FPR	FNR	UPR
nDSM	94,49	10,69	5,51	0,00
AdaBoost 3F	87,44	1,33	7,31	5,25
AdaBoost 5F	91,17	3,95	7,08	1,75
AdaBoost 7F	88,84	1,57	4,76	6,40

Classifier	OA	R	TUR
nDSM	91,24	83,95	0,00
AdaBoost 3F	96,13	97,50	8,16
AdaBoost 5F	94,66	93,18	4,30
AdaBoost 7F	96,97	97,10	8,96

Table 1: Results of pixel-based classification using different sets of features and metrics

AdaBoost 3F, 5F and 7F differ for the set of features; 3F classifier uses Δh, Δp and NDVI, 5F adds Green and Blue; AdaBoost 7F classifies data using all features (excluding GNDVI). The AdaBoost 3F guarantees the best performance if compared with AdaBoost 5F/7F; adding more features other than Δh, Δp and NDVI, the classifier misclassifies data due lack of spectral separability (confirmed by ReliefF). All the classified data are also used for the road extraction; in particular the binary image obtained by considering land (bit set to one) and remaining classes (bit set zero) represents the input for roundabout and road extraction; the approach and results are presented in the following section.

[1] TP/FP = true/false positive TN/FN = true/false negative UP/UN = unclassified positive/negative

4 ROAD EXTRACTION

In this section we present preliminary results on road/roundabout extraction starting from classified data; the proposed approach works fine when the area is urban; modern cities often grows around main ancient perpendicular roads (cardus-decumanus). The key idea behind the algorithm is the "line growing"; more details about algorithm are discussed in next sub-sections.

4.1 Filtering

Filtering is a preliminary process before road extraction; this activity is necessary for two main reasons: the first one is the presence of noisy classified data, because pixel-based classification suffers of noise; other approaches based on regions (object-based classification) can reduce it. The second problem that influences the quality of road extraction is the presence of trees/canopies; the chosen approach is a non-linear filter; if pixels that appertain to tree class have neighbours classified as "land", then they are assigned to land class. The advantage of using this filter, is the reduction of effect produced by occlusions. In Fig. 7 the result of the filtering process is shown.

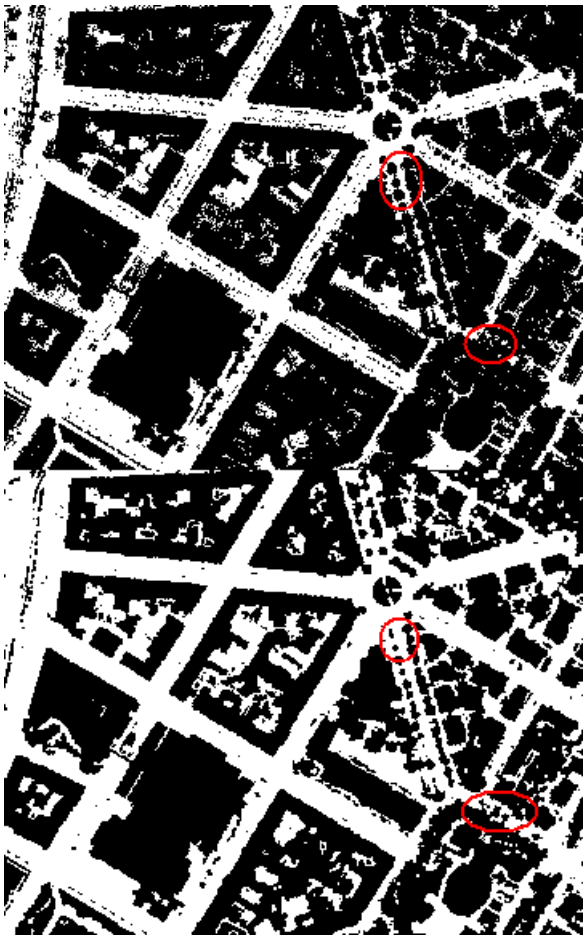


Figure 7: Filtering. In the top image white pixels are classified as land; classification is noisy due to the presence of small objects as vehicles; in the bottom, the non-linear filter allows to reduce significantly the effect of noise and occlusions

Non-linear filter consists of two steps: the first one is the reduction of noise using morphological operators. We applied three algorithms: *opening* to remove small objects, morphological *reconstruction* to retrieve boundaries and *closing* to fill small holes; the structuring element used was disk of size two. Second step is

to reduce the effect of canopy occlusions. The non-linear filter is a moving kernel of 7x7 that substitutes pixels classified as "tree" if and only if neighbours are "land". In Fig. 7 red blobs put in evidence the reduction of occlusions due to the presence of trees.

4.2 Roundabout Extraction

After filtering, before extract roads, roundabouts are identified using a Hough transform applied to circular shapes. Hough transform is useful to extract well-defined shapes as lines, circles or ellipse; the major drawback is the computational time, which is high especially for complex shapes (in terms of number of parameters) as ellipses. In Fig. 8, a roundabout extracted from Mannheim dataset is shown.

Figure 8: Hough transform applied to Mannheim dataset to find circular shapes as roundabouts

The Hough transform usually tends to overfit the real number of circular shapes; we use a double thresholding (min - max) to filter the output of Hough. Roundabout shown in Fig. 8 is centred on $x = 1194$, $y = 378$ with a radius of 47 pixels (about 22m); min-max values are determined from typical values for small and/or large roundabouts. The input image for the Hough transform is obtained by the classified data; in Fig.7 the binary image is shown; the approach was tested also on different images to validate the extraction procedure; it is also possible to extract more complex roundabouts (e.g., elliptical) using the Randomized Hough Transform also in presence of partial occlusions (Hahn et al., 2007). The roundabouts identified with Hough transform mask the filtered data supporting the next step: line extraction and clustering.

4.3 Linear Road Extraction

Segment extraction approach starts from the filtered data masked with roundabouts. Proposed method is similar to region growing technique usually applied in image segmentation; starting from a seed point of size one, classified as "land" the algorithm expand regions (in this case a segment) adding one or more pixels of same class; growing process ends when the region meets a set of N pixels classified as not-land. The main difference with the classical region growing is the size of growing space. In the case of image segmentation, growing space is 2D; in the case examined in this paper, the expansion is one-dimensional; next pixel (in both direction left and right) is calculated using the line parameters in terms of angular value; the pseudo-code of proposed algorithm is shown in Algorithm 1. The algorithm has two parameters: $T1$ and $T2$. $T1$ is used to stop growing process if $T1$ consecutive points (spurious pixels) classified as not-land are encountered. $T2$ is a criteria to specify the minimum length of segment; the values of these parameters were set to $T1 = 2$ and $T2 = 30$; a pyramidal down-scaling (factor 0.5) is performed on filtered data to reduce the complexity of computation. The $calculate\_segment\_points(j, i, \theta)$ function, given an origin (j, i) in the image reference system, and an orientation θ, returns a list of pixels that belongs to the parametrized line, while the $count\_spurious\_pixels(s, start, x)$ returns the number of spurious pixels (classified as not land) along the segment. The add function adds a segment to vector S or adds a pixel $p[k]$ to the vector of candidate pixels belonging to a segment. In Fig.9 an example of segment extraction on a synthetic image is shown; the best segment orientation is chosen as the angular value that minimizes the number of segments extracted; if thresholds $T1$ and $T2$ are set properly, the minimum point is not strongly afflicted by the presence of noise.

Algorithm 1 Extraction of linear segments

Require: x vector of classified data
S vector of extracted segments
s vector of candidate pixels belonging to a segment
p vector of aligned pixels
for $j = 0$ to $j < height$ **do**
  for $i = 0$ to $i < width$ **do**
    for $\theta = -\pi/2$ to $\pi/2$ **do**
      $p \leftarrow calculate\_segment\_points(i, j, \theta)$
      $start \leftarrow 0$
      $s.clear$
      for $k = 0$ to $k < p.size$ **do**
        $n = count\_spurious\_pixels(s, start, x)$
        if $n > T2 \vee i == (width - 1) \vee j == (height - 1)$ **then**
          if $p.length > T1$ **then**
            S.add(s)
            s.clear
            $start \leftarrow k + 1$
          else
            $s.add(p[k])$
          end if
        end if
        $k \leftarrow k + 1$
      end for
      $\theta \leftarrow \theta + 1$
    end for
    $i \leftarrow i + 1$
  end for
  $j \leftarrow j + 1$
end for

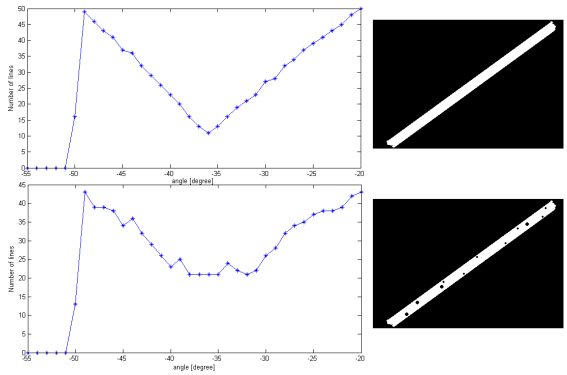


Figure 9: Segment extraction. Top image represent an ideal segment extraction while in the bottom it is tested a noisy image

The best orientation for a road is chosen by minimizing the number of extracted segments (as shown in Fig.9); a road can be defined as the minimum set of segments with a length greater than $T2$ and same angular value. The set of segments which forms a road is created applying a clustering algorithm; the DBSCAN (Ester et al., 1996) is adopted to group the set of extracted segments. A segment belongs to a cluster if and only if the distance between the initial point of segment and the nearest neighbour is under a threshold; if this geometric criteria is satisfied the lengths of clusterized segments are also checked. If the length are comparable (in terms of distance from the mean value of the cluster) the set of cluster is labelled as road and the centerline is calculated. In Fig.10 a series of tests on Mannheim data-set for different orientations is shown. Tests put in evidence that the algorithm, owing to the clustering, does not consider incoherent segments (Fig.10c).

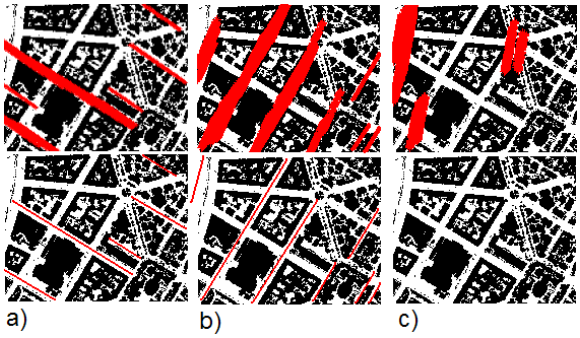


Figure 10: Road extraction for three different angles; segments are the thick red lines (bottom), while raw ones are shown in top

The extracted geo-referenced and vectorial road graph with the proposed technique is shown in Fig.11; some roads are not correctly identified due to presence of high density canopies.

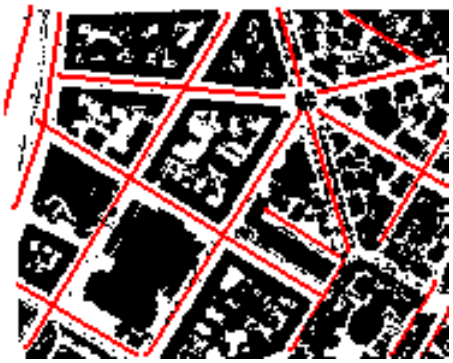


Figure 11: Road graph for Mannheim data-set

5 CONCLUSIONS AND FUTURE WORK

In this paper, we presented a complete methodology to solve the problem of automatic extraction of urban objects from multisource aerial data. The procedure, which consists of sequential steps, takes advantage of classified data with a powerful machine learning algorithm as AdaBoost with CART as weak learner. The capability of distinguishing among four classes in an urban area as Mannheim increases the set of possible applications; two test cases were presented: building and road extraction. In the case of building extraction, the fusion of spectral data with LiDAR data using AdaBoost overtakes the limits of a simple nDSM thresholding especially when canopies have a high density. The proposed road extraction method allows to reduce the effect of occlusions;roads, extracted with the "line growing" approach enhanced with clustering, well match with a photo-interpretation process. As future works, more tests on more complex data with curved lines will be performed; moreover different weak learners based on RBF Neural Networks will be tested.

ACKNOWLEDGMENT

We would like to thank C. Nardinocchi and K. Khoshelham at Delft University of Technology for their helpful support.

REFERENCES

Bacher, U. and Mayer, H., 2005. Automatic road extraction from multispectral high resolution satellite images. Proceedings of CMRT05.

Breiman, L., Friedman, J., Olshen, R. and Stone, C., 1984. Classification and Regression Trees. Wadsworth and Brooks, Monterey, CA.

Clode, S. P., Rottensteiner, F. and Kootsookos, P., 2005. Data acquisition for 3d city models from lidar extracting buildings and roads. Proceedings of CMRT05.

Duda, R. O., Hart, P. E. and Stork, D. G., 2000. Pattern Classification. Wiley-Interscience Publication.

Ester, M., peter Kriegel, H., S, J. and Xu, X., 1996. A density-based algorithm for discovering clusters in large spatial databases with noise. In: Proceedings of the Knowledge Discovery and Data Mining Conference, AAAI Press, pp. 226–231.

Frontoni, E., Khoshelham, K., Nardinocchi, C., Nedkov, S. and Zingaretti, P., 2008. Comparative analysis of automatic approaches to building detection from multisource aerial data. Proceedings of GEOBIA.

Guo, Y., Bai, Z., Li, Y. and Liu, Y., 2007. Genetic algorithm and region growing based road detection in sar images. Proceedings of Int. Conference on Natural Computation pp. 330–334.

Hahn, K., Han, Y. and Hahn, H., 2007. Extraction of partially occluded elliptical objects by modified randomized hough transform. In: Proceedings German conference on Advances in Artificial Intelligence, Springer-Verlag, pp. 323–336.

Kass, M., Witkin, A. and Terzopoulos, D., 1987. Snakes: Active contour models. Int. J. of Computer Vision 1(4), pp. 321–331.

Liu, H. and Motoda, H., 2008. Computational Methods of Feature Selection. Chapman and Hall/CRC.

Lu, Y., Trinder, J. and Kubik, K., 2006. Automatic building detection using the dempster-shafer algorithm. Photogrammetric Engineering and Remote Sensing 72(4), pp. 395–403.

Marikhu, R., Dailey, M. N., Makhanov, S. and Honda, K., 2006. A family of quadratic snakes for road extraction. Lecture Notes in Computer Science ACCV 2007 4843, pp. 85–94.

Mena, J., 2003. State of the art on automatic road extraction for gis update: a novel classification. Pattern Recognition Letters 24, pp. 3037–3058.

Schapire, R. and Singer, Y., 1999. Improved boosting algorithms using confidence-rated predictions. Machine Learning 37(3), pp. 297–336.

Walter, V., 2004. Object-based classification of remote sensing data for change detection. ISPRS Journal of Photogrammetry and Remote Sensing 58, pp. 225–238.

Weidner, U. and Forstner, W., 1995. Towards automatic building extraction from high resolution digital elevation models. ISPRS J. of Photogrammetry and Remote Sensing 50(4), pp. 38–49.

ROAD ROUNDABOUT EXTRACTION FROM VERY HIGH RESOLUTION AERIAL IMAGERY

M. Ravanbakhsh, C. S. Fraser

Cooperative Research Center for Spatial Information, Department of Geomatics
University of Melbourne VIC 3010, Australia
[m.ravanbakhsh, c.fraser]@unimelb.edu.au

KEY WORDS: roundabout detection, feature extraction, topographic database, high resolution imagery, snake model, level sets

ABSTRACT:

Road roundabouts, as a class of road junctions, are generally not explicitly modelled in existing road extraction approaches. This paper presents a new approach for the automatic extraction of roundabouts from aerial imagery through the use of prior knowledge from an existing topographic database. The proposed snake-based approach makes use of ziplock snakes. The external force of the ziplock snake, which is a combination of the Gradient Vector Flow force and the Balloon force, is modified based on the shape of the roundabout central island to enable the roundabout border to be delineated. Fixed boundary conditions for the proposed snake are provided by the existing road arms. A level set framework employing a hybrid evolution strategy is then exploited to extract the central island. Black-and-white aerial images of 0.1 m ground resolution taken over suburban and rural areas have been used in experimental tests which have demonstrated the validity of the proposed approach.

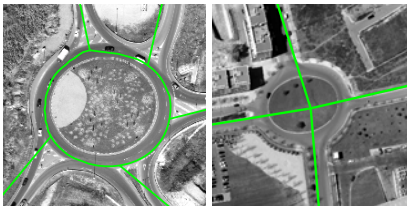
1. INTRODUCTION

The need for accurate spatial databases and their automatic updating is increasing rapidly. Road networks form key information layers in topographic databases since they are used in such a wide variety of applications. As the extraction of roads from images is still generally manual, costly and time-consuming, there is a growing imperative to automate the process. However, such a feature extraction task has longed proved difficult to automate. The problem for automatic road extraction lies mostly in the complex content of aerial images. To ease the complexity of the image interpretation task, prior information can be used (Gerke, 2006; Boichis et al., 2000; Boichis et al., 1998; De Gunst, 1996). This often includes the provision of data from an external topographic database.

Roundabouts, as a class of road junctions, are important components of a road network and if modelled well can improve the quality of road network extraction (Boichis et al., 1998). However, there are only few approaches which are dedicated to this task. Boichis et al. (2000) presented a knowledge based system for the extraction of road junctions and roundabouts. The method assumed that the description of simple road junctions and roundabouts is the same in the external database, so a previous detector has to certify the presence of the circular form. A parametric Hough Transform is used for this purpose. The roundabout is reconstructed after straight parts of the connecting roads, curved parts including splitter islands, and the circulating road are extracted.

These elements are connected using geometric and radiometric continuities. In the approach, roads are treated as linear objects. Thus, elements such as the central island and the roundabout outline are not extracted, so kind of modelling does not always reflect the required degree of detail. In Fig. 1, vector data is superimposed on sample images to illustrate the problem. The image resolution is such that the roundabout's central area covers the central island and the circulating roadway. In Fig. 1b, the roundabout is represented as point object neglecting the central island and the circulating roadway. Thus, a detailed modelling of roundabouts is needed for data acquisition purposes at large scales.

The detailed modelling of road roundabouts area objects is discussed in this paper, and an approach for their automatic extraction is proposed. This uses an existing topographic database leading to the extraction of refined roundabout data. In the following section, a model for roundabouts is first introduced. The stages of the proposed strategy are then illustrated in Sect. 3. Results from the implementation of the proposed approach using aerial imagery of 0.1 m ground resolution are presented and discussed in Sect. 4, together with an evaluation of their quality. Finally concluding remarks are offered.



(a) (b)

Figure 1. Superimposition of vector data on high resolution aerial images of road roundabouts.

2. ROUNDABOUT MODEL

Illustrated in Fig. 2a is the conceptual two-part model of a roundabout, the parts being the roundabout itself and the road arms. The roundabout, where road arms are connected, is in turn composed of the roundabout border and its central area where a central island is located. A road arm is a rectilinear object which is represented as a ribbon with a constant width and two parallel road edges. Disturbances such as occlusions and shadows are not explicitly included in the model.

3. EXTRACTION APPROACH

The proposed strategy consists of three steps (Fig. 2b). First, the topographic geospatial database is analysed and different types

of parameters for roundabouts are derived. Second, the central island is extracted using a level set approach making use of prior information obtained from the previous step. Finally, the roundabout is reconstructed using a snake-based method. The proposed approach has the aerial image, a topographic database and the road arms as input, and the roundabout border connected to the existing road arms as output. The reader is referred to Ravanbakhsh et al. (2008) or Ravanbakhsh (2008) for a description of how the road arms are extracted.

3.1 Pre-analysis of topographic database

Roundabouts are usually represented in topographic databases in one of two ways, either as an area object when the diameter of the inscribed circle is larger than the threshold (Fig. 1a), or as a point object when the diameter of the inscribed circle is small (Fig. 1b). The actual representation threshold varies in different topographic databases. This vector data is used to focus the extraction process to the image regions where roundabouts are located. Furthermore, the approximate diameter of the central island and width of the circular roadway can be initially determined (Fig. 3).

It is noteworthy that the width of the circular roadway depends mainly upon the number of entry lanes. The width of entry lanes is also derived from vector data. According to construction standards, the roadway must be at least as wide as the maximum entry width and generally should not exceed 1.2 times this width (U.S. Federal Highway Administration, 2000). In case that a roundabout appears as a point object, attributive information must be included in the topographic database implying that the node represents a small roundabout. This means that the diameter of the inscribed circle is below the threshold that has been defined in the topographic database.

3.2 Central island extraction

With roundabouts, a correct extraction of the central island helps facilitate the extraction of the outline. The reason for this is that the central island, when enlarged, influences the shape of the roundabout outline. The initial detection of the central island can then provide a good idea of how the outline appears in the image. The proposed method to detect central islands is based on level sets.

Geometric active contours were introduced by Caselles et al. (1993) and Malladi et al. (1995). These models are based on curve evolution theory and the level set method. The basic idea is both to represent contours as the zero level set of an implicit function in a higher dimension, usually referred to as the level set function ϕ, and to evolve the level set function according to a partial differential equation (PDE). It is well known that a signed distance function, a function which introduces the minimum distance from every point in a defined domain to the zero isocontour of a level set function, must satisfy the desirable property of $|\nabla\phi|=1$ (Osher and Fedkiw, 2002). The following formula has been proposed to provide the internal energy of a snake which penalizes the deviation of ϕ via a signed distance function (Li et al., 2005):

$$P(\phi) = \int_{\Omega} \frac{1}{2}(|\nabla\phi| - 1)^2 \, dxdy \tag{1}$$

$P(\phi)$ is a metric to characterize how close the function ϕ is to a signed distance function in a specified computational domain $\Omega \subset R^2$. The external energy is defined by

$$E_m(\phi) = \lambda L_g(\phi) + v A_g(\phi) \tag{2}$$

where λ>0 and v is a constant and the length term $L_g(\phi)$ and area term $A_g(\phi)$ are defined by

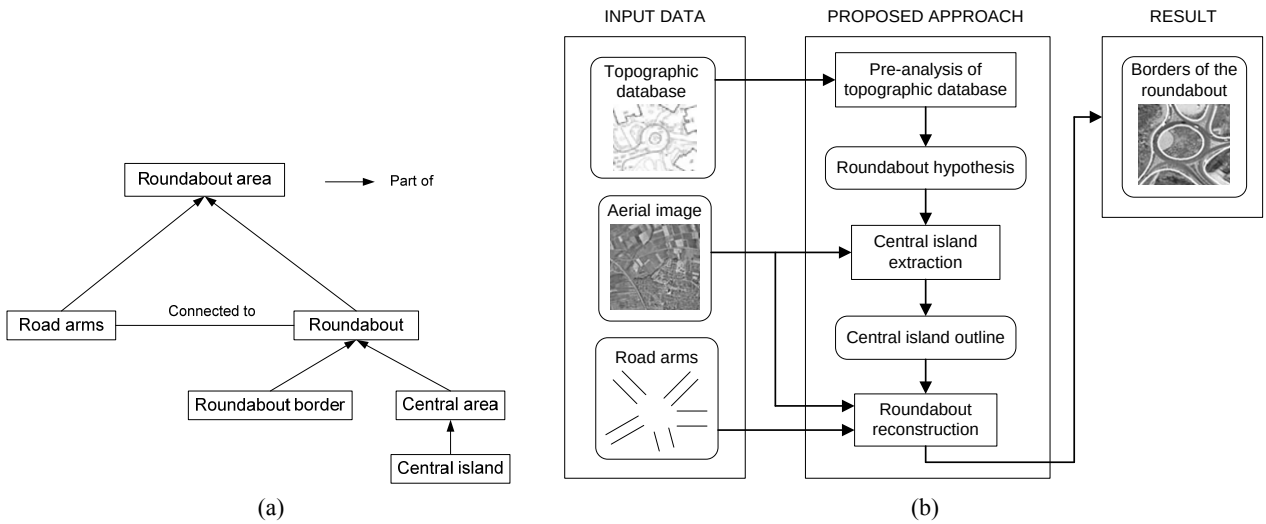
$$L_g(\phi) = \int_{\Omega} g\delta(\phi) |\nabla\phi| \, dxdy \tag{3}$$

$$A_g(\phi) = \int_{\Omega} gH(-\phi) dxdy \tag{4}$$

with the *edge indicator function* g being given by

$$g = \frac{1}{1 + |\nabla G_\sigma * I|^2} \tag{5}$$

Here, H is the Heaviside function, δ the univariate Dirac function, G_σ the Gaussian kernel with standard deviation σ, and I image intensity.



(a) (b)

Figure 2. (a) Roundabout model and (b) workflow of roundabout extraction.

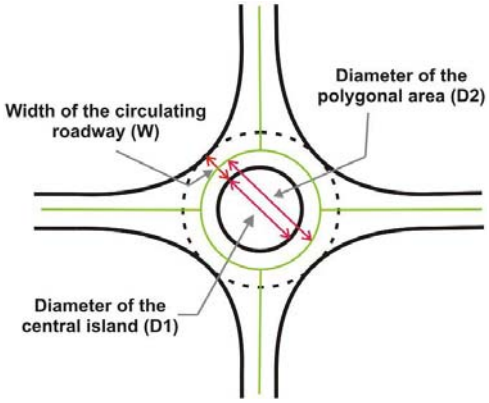


Figure 3. Schematic illustration of the relationship between roundabout geometric parameters. Vector data is in green.

The resulting total energy function can now be defined:

$$E(\phi) = \mu P(\phi) + E_m(\phi) \tag{6}$$

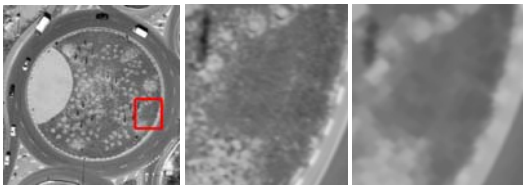
where μ>0 controls the balance between the first and second term. The evolution equation of the level set function is then obtained via calculus of variation (Courant & Hilbert, 1953) and application of the steepest descent process for minimization of the energy functional equation (Li et al., 2005) as

$$\frac{\partial \phi}{\partial t} = \mu[\Delta\phi - \mathrm{div}(\frac{\nabla\phi}{|\nabla\phi|})] + \lambda\delta(\phi)\mathrm{div}(g\frac{\nabla\phi}{|\nabla\phi|}) + \mathrm{vg}\delta(\phi) \tag{7}$$

For all examples of central island detection, the same set of control parameters, λ=4, μ=0.13, v=2 and the time step ∂t=2, were tuned for the evolution equation (Eq. 7).

Since either evolution type alone, shrinking and expansion, has its own limitations, a hybrid evolution strategy is employed. For instance, in case of only shrinking curve evolution, vehicles on the circulating roadway, and in case of only expansion curve evolution, structures inside the central island, can block the motion of the curve toward the central island's border. Using a hybrid evolution strategy overcomes various kinds of disturbances often present inside and outside the central island.

Often before the curve evolution begins, a pre-processing step is necessary to remove some fine features that might hinder the curve motion. First, a morphological closing operator is applied in order to remove dark spots and subsequently the opening with the same structuring element (disk structuring element; size=2) is performed to eliminate small bright features followed by Gaussian smoothing (Fig. 4c).

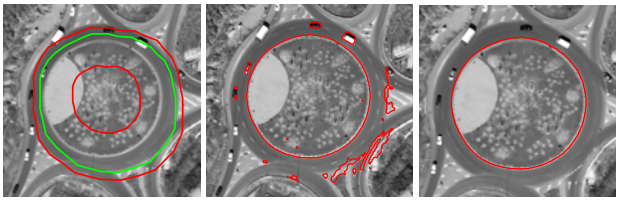


(a) (b) (c)

Figure 4. Pre-processing sequence: (a) Original image, (b) cut-out marked by the red box in (a), and (c) pre-processed result.

Shown in Fig. 5 is the first sequence for island extraction, when shrinkage curve evolution is applied. After the vertices of the polygonal area identified as a roundabout object in the topographic database are located, the polygon is enlarged so that its increased area is one-tenth more than its initial area (Fig. 5a), thereby making sure that the new polygon is located outside the central island. Subsequently, shrinkage evolution begins through use of level sets. Among the obtained closed curves in Fig. 5b, the one with the largest area is selected as the initial guess for the island (Fig. 5c). This island candidate is subject to further processing.

Next, the initial polygon obtained from vector data is made smaller so that its area is reduced by half (Fig. 5a). Subsequently, expansion curve evolution occurs (Fig. 6a). The largest resulting closed curve is most likely the desired solution due to the fact that the island is the largest object within the computational domain. This closed curve, however, can often not be regarded as the island because many disturbing features such as trees and various structures exist inside the island. This can block the motion of the evolving curve towards the island boundary. Leakages are therefore created at some points along the boundary of disturbing features where zero level curves have stopped in order to pass over them.



(a) (b) (c)

Figure 5. First sequence for island extraction: (a) Polygonal vector data (green) and its enlarged and reduced forms(red), (b) shrinking curve evolution result after 1335 iterations, and (c) the eventual result of shrinking evolution.

With the assumption that disturbing objects inside the island do not contain smooth boundaries, cubic spline approximation is carried out to provide leakages (Fig. 6b). Subsequently, expansion evolution and spline approximation are repeatedly carried out (Fig. 6c) until no change in the position of the curve is reported. Again, the largest closed curve is regard as the island (Fig. 6d). Now that the results of island detection from the iterative expansion and shrinkage curve evolution have been obtained, the image positions of the resulting curves are compared and those points which are close to each other are selected, thereby eliminating curve positions that are not located on the island boundaries. The selection of points is based on their closeness in such a way that points having a distance below a certain threshold are selected. The final result is obtained when an ellipse is fitted to the selected points.

When a roundabout appears as a point object in the topographic database (Fig. 1b), the same hybrid evolution strategy is used but with a different initialization because the diameter of the inscribed circle is known to be below a given threshold, but how small it is is unknown. This brings some limitations for the shrinkage curve evolution. In order to apply the shrinkage evolution, the initial zero level curve must be placed outside the island. Since the approximate diameter of the inscribed circle is unknown, three successive circles are defined (Fig. 7a), on each of which the shrinkage curve evolution is carried out separately. The diameter of the circle interior to the central island is 10 m and the diameters of exterior circles have an interval of 3 m. The results of shrinkage evolution on each initial curve from the largest to the smallest circle are depicted in Figs. 7b, c and d. In the next step, the iterative expansion evolution is carried out

similarly to the method described earlier with the exception that the initial curve is defined as a circle around the roundabout node so that it must be placed inside the island (Fig. 7a).

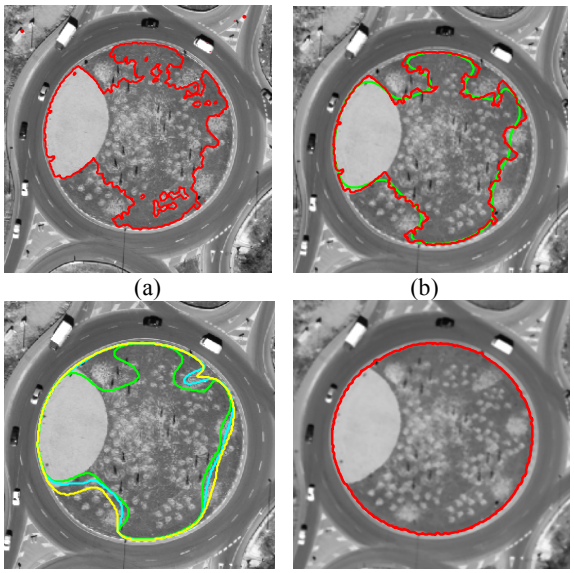


Figure 6. Island extraction: (a) Expansion evolution result after 1330 iterations, (b) selected curve (red) and approximated cubic spline (green), (c) other curves resulting from iterative curve evolution, and (d) eventual result of expansion evolution.

The diameter of the circle needs to be less than the threshold which dictates whether islands are regarded as point or area objects in the topographic database. By experiment, it is safer to define a circle with a diameter as one-third of this threshold. The expansion result is compared with each group of shrinkage results separately, and points that are close enough to each other are selected. These points are candidates for ellipse fitting. The fitting result for a case with the highest number of points is more likely to produce a correct result of island extraction (Fig. 7f).

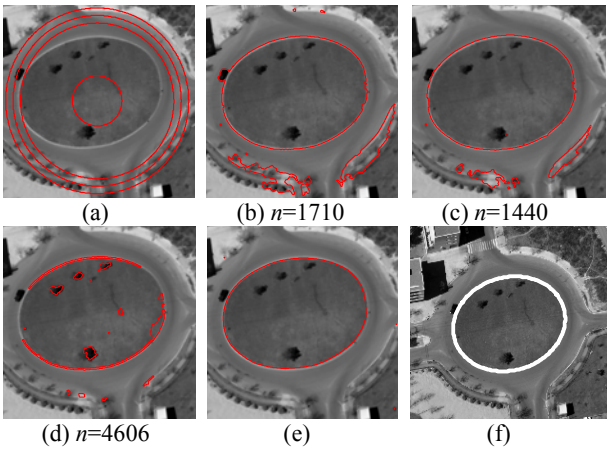


Figure 7. First sequence for island extraction: (a) initial successive circles, 3 located outside central island for shrinking evolution and 1 inside for expansion curve evolution; (b), (c) and (d) results of the shrinking evolution for exterior circles from large to small (*n* denotes iteration number); (e) result of iterative expansion evolution to interior circle; (f) final result.

Extracted central islands are verified using the existing information derived from the topographic database. When a roundabout appears in the database as an area object, as shown in Fig. 3, the diameter of its central island (D1) obtained from the extraction process must only differ from that obtained from vector data (D2) by a small amount. In an ideal situation, the difference (ΔD) corresponds to the width of the circulating roadway (W), i.e. $W=\Delta D$. In practice, due to the imprecise digitization of roundabouts, polygonal vector data do not always lie on the middle axis of the circulating roadway, but somewhere within its area. Therefore, ΔD is expected to be within the range of 0 to 2W, i.e. $0< \Delta D<2W$.

In the case where a roundabout appears as a point feature, the diameter of the extracted central island must fall within a predefined range whose highest value is the threshold below which a roundabout is regarded as a point object and whose lowest value is the minimum possible diameter for a central island.

3.3 The Snake Model for Roundabout reconstruction

The snake model, or parametric active contour method (Kass et al., 1988), used to delineate the roundabout outline is now briefly overviewed to provide a basis for further discussion. Further details are provided in Ravanbakhsh et al. (2008) and Ravanbakhsh (2008). Snakes are especially useful for delineating objects that are hard to model with rigid geometric primitives. They are thus well suited to modeling roundabouts since the borders are of diverse shape with various degrees of curvature. Snakes are polygonal curves associated with an objective function that combines an image term (external energy) and measurement of the image force (e.g. the edge strength). There is also a regularization term (internal energy) and a minimization of the tension and curvature of the polygon. The curve is deformed so as to iteratively optimize the objective function. Traditional snakes are sensitive to noise and need precise initialization. Since roundabout borders have various degrees of curvature, a close initialization cannot often be provided. As a result, traditional snakes can easily get stuck in an undesirable local minimum.

To overcome these limitations, the *ziplock snake* model was developed (Neuenschwander et al., 1997). A ziplock snake consists of two parts: an active part and a passive part. The active part is further divided into two segments, indicated as head and tail, respectively (Fig. 8). The active and passive parts of the ziplock snake are separated by moving force boundaries. Unlike the procedure for a traditional snake, the external force derived from the image is turned on only for the active parts. Thus, the movement of passive vertices is not affected by any image forces. Starting from two short pieces, the active part is iteratively optimized to image features, and the force boundaries are progressively moved towards the centre of the snake. Each time that the force boundaries are moved, the passive part is re-interpolated using the position and direction of the end vertices of the two active segments. Optimization is stopped when force boundaries meet each other.

Ziplock snakes need far less initialization effort and are less affected by the shrinking effect from the internal energy term. Furthermore, their computation is more robust because the active part, whose energy is minimized, is always quite close to the contour being extracted. This modified snake model can detect image features even when the initialisation is far away from the solution. However, it can still become confused in the presence of disturbances. In high resolution aerial images, such

disturbances may destabilize the ziplock's active vertices. As a result convergence may not occur or the snake may get trapped near the initial position. As a means of overcoming this problem, an external force with a large capture range can be applied.

The *Gradient Vector Flow* (GVF) field (Xu & Prince, 1997), which is an example for such an external force, is used in the proposed approach. The GVF field was aimed at addressing two issues: a poor convergence to concave regions, and problems associated with the initialisation. It is computed as a spatial diffusion of the gradient of an edge map derived from the image. This computation causes diffuse forces to exist far from the object, and crisp force vectors to be near the edges. The GVF field points toward the object boundary when very near to the boundary, but varies smoothly over homogeneous image regions, extending to the image border. The main advantage of the GVF field is that it can capture a snake from a long range. Thus, the problem of far initialization can be alleviated.

The Evolution of a ziplock snake is illustrated in Fig. 8. The snake is fixed at the head and tail, and it consists of two parts, the active and the passive vertices. These parts are separated by moving force boundaries. The active parts of the snake consist of the head and tail segments.

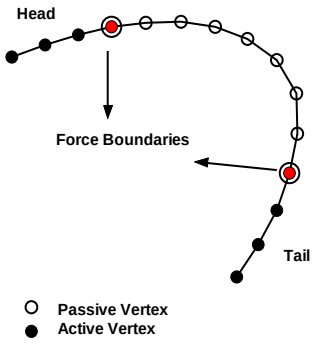


Figure 8. Evolution of a ziplock snake.

The GVF is defined to be the vector field $G(x,y) = (u(x,y), v(x,y))$ that minimizes the energy functional:

$$E = \iint \mu(u_x{}^2 + u_y{}^2 + v_x{}^2 + v_y{}^2) + |\nabla f|^2 |G - \nabla f|^2 \, dxdy \tag{8}$$

where ∇f is the vector field computed from $f(x,y)$ having vectors pointing toward the edges. $f(x,y)$ is derived from the image and it has the property that it is larger near the image edges.

The regularization parameter μ should be set according to the amount of noise present in the image; more noise requires a higher value of μ. Through use of calculus of variations (Courant & Hilbert, 1953), the GVF can be found by solving the following Euler equations:

$$\begin{aligned} \mu\nabla^2 u - (u - f_x)(f_x{}^2 + f_y{}^2) = 0 \\ \mu\nabla^2 v - (v - f_y)(f_x{}^2 + f_y{}^2) = 0 \end{aligned} \tag{9}$$

where ∇^2 is the Laplacian operator and f_x and f_y are partial derivatives of f with respect to *x* and *y*.

Let $V(s) = (x(s), y(s))$ be a parametric active contour in which s is the curve length and *x* and *y* are the image coordinates of the 2D-curve. The internal snake energy is then defined as

$$E_{\text{int}}(V(s)) = \frac{1}{2}\left[\alpha(s)\,|V_s(s)|^2 + \beta(s)\,|V_{ss}(s)|^2\right] \tag{10}$$

where V_s and V_{ss} are the first and second derivatives of V with respect to s. The functions $\alpha(s)$ and $\beta(s)$ control the elasticity and the rigidity of the contour, respectively. The global energy

$$E = E_{\text{int}}(V(s)) + E_{img}(V(s)) \tag{11}$$

needs to be minimized, with $\alpha(s) = \alpha$ and $\beta(s) = \beta$ being constants. Minimization of the energy function of Eq. 11 gives rise to the following Euler equations:

$$-\alpha V_{ss}(s) + \beta V_{ssss}(s) + \frac{\partial E_{img}(V(s))}{\partial V(s)} = 0 \tag{12}$$

where $V(s)$ stands for either $x(s)$ or $y(s)$, and V_{ss} and V_{ssss} denote the second and fourth derivatives of V, respectively. After approximation of the derivatives with finite differences, and conversion to vector notation with $V_i = (x_i, y_i)$, the Euler equations take the form

$$\begin{aligned} &\alpha_i(V_i - V_{i-1}) - \alpha_{i+1}(V_{i+1} - V_i) + \beta_{i-1}\left[V_{i-2} - 2V_{i-1} + V_i\right] \\ &-2\beta_i\left[V_{i-1} - 2V_i + V_{i+1}\right] + \beta_{i+1}\left[V_i - 2V_{i+1} + V_{i+2}\right] + G(u,v) = 0 \end{aligned} \tag{13}$$

where $G(u,v)$ is the GVF vector field . Eq. 13 can be written in matrix form as

$$KV + G(u,v) = 0 \tag{14}$$

where K is a pentadiagonal matrix.

Finally, the motion of the GVF ziplock snake can be written in the form (Kass et al., 1988)

$$V^{[t]} = (K + \gamma\, I)^{-1} * (\gamma\, V^{[t-1]} - \kappa\, \mathrm{G(u,v)}\,|_{v^{[t-1]}}) \tag{15}$$

where γ stands for the viscosity factor (step size) determining the rate of convergence and t is the iteration index. κ alters the strength of the external force.

It is noteworthy that the proposed model still might fail to detect the correct boundaries in the following cases:

- High variation of curvature at the roundabout border resulting in an initialization that is too poor in some parts, with the consequence that the snakes becomes and remain straight.
- The roundabout central area lacks sufficient contrast with the surroundings, causing the curve to converge to nearby features.

Through the use of shape description parameters such as curvature computed from the snake vertices, another force can be added to the GVF force field. This is the so-called *balloon force*, which lets the contour have a more dynamic behaviour (Cohen, 1991), thereby addressing the two described problems. This new force, which makes the contour act like a balloon, applies an inflating effect to the contour to localize the concave part of the roundabout outline:

$$F = k_1 \vec{n}(s) \tag{16}$$

where $\vec{n}(s)$ is the normal unitary vector of the curve at point $V(s)$ and k_1 is the amplitude of the force. The combination of the GVF force field and the balloon force modifies Eq. 15 to the form

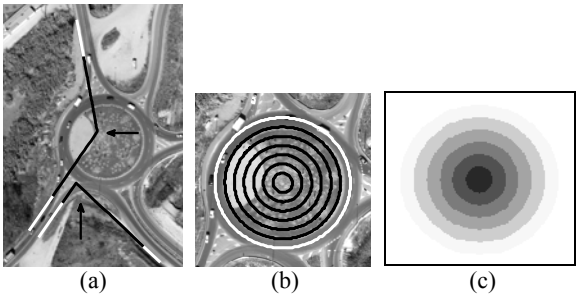
$$V^{[t]} = (K + \gamma\, I)^{-1} * (\gamma\, V^{[t-1]} - \kappa\, G(u,v)_v\,|_{v^{[t-1]}} - k_1 \vec{n}(s)) \tag{17}$$

The balloon force is activated when the snake's passive and active parts are approximately straight, i.e. their overall curvature, which is defined as the sum of the absolute curvatures along the curve, is below a threshold. It is applied only on the passive part of the curve. This is regarded as lying outside the roundabout's border, whereas the snake at the active parts is assumed to be on the right track. The direction in which the balloon force is applied is towards the roundabout central area. However, in order to be able to delineate the roundabout outline, the balloon force has to be applied in two different directions, central island inwards and outwards (Fig. 9a).

The answer to the question of when and in which direction the balloon force needs to be applied differs for different samples. As a result, several parameters need to be tuned on an ad hoc basis to address this question, which is not a desirable requirement. To resolve this, the external force field of the snake approach described so far is modified based on the shape of the central island. As the shape of the roundabout outline corresponds to the shape of the enlarged central island, the island is enlarged to an extent depending on the width of the circulating roadway (Fig. 9b). Subsequently the snake external force field is modified based on the enlarged central island. The external force field in the enlarged central island is replaced with the GVF of an intensity-step image (Fig. 9c) whose main characteristic is that its external force points directly from the centre outwards so that snakes situated in this area are drawn toward the outline of the roundabout.

The intensity-step image is generated from a signed distance function. To generate this function, the border of the enlarged central island is taken as the reference (Fig. 9b). Successive concentric layers at a specific distance interval from the reference to the centre point are then defined. Conversely, proportional to the distance of each layer to the reference, an intensity value is calculated and assigned to the respective layer, i.e. layers closer to the reference curve are brighter and vice versa.

The obtained intensity-step image has a gradual increase of intensity values from the centre point towards the reference curve. Consequently, its GVF field points directly outward. The modified force field pulls the snakes toward the outline even if the initialization is far away from true borders. Furthermore, with this modified force field, problems created by the presence of various kinds of disturbances such as trees and vehicles within and outside the central island are overcome. An example illustrating the improved result using the proposed modified force field is shown in Fig. 10. The complete reconstruction of a roundabout using the proposed modified snake model is shown in Fig. 11, along with intermediate results.



(a) (b) (c)

Figure 9. (a) Two directions in which the balloon force is applied; (b) reference for the signed distance function (white curve) computation and concentric regions (black curves); (c) intensity-step image from the signed distance function.

(a) (b) (c)

Figure 10. The effect of the modified external force field: (a) intersection lines (black) from initial snakes, (b) results from unmodified GVF, and (c) improved results with modified GVF.

4. RESULTS AND EVALUATION

The proposed approach was tested using 0.1m GSD panchromatic aerial orthoimagery covering rural and suburban areas. The Authoritative Topographic Cartographic Information System of Germany (ATKIS), which nominally corresponds to a mapping scale of 1:25,000, was used as the source of external vector data. Roads are modelled as linear objects in ATKIS. Tests were conducted on 10 roundabouts. Sample results that highlight the capabilities of the proposed approach are shown in Fig. 12, where is can be seen that the method can deal with a variety of disturbances inside and outside the central island. Also, most of the roundabout borders were captured correctly. However, in areas where the curvature of the outline was too high, as is the case in the top-left example (lower border) and top-right image (right border), roundabout borders were extracted with some deviation.

In order to evaluate the performance of the approach, the extracted roundabout areas were compared to the manually plotted roundabouts used as reference data. The comparison was carried out by matching the extracted road borders resulting from the connection of the roundabout to its associated road arms to the reference data using the so-called *buffer method* (Heipke et al. 1998). Although the buffer width can be defined using the required accuracy of ATKIS, which for a road object is defined as 3m, it was decided to set the buffer width within the range of 0.5 m to 3 m, i.e. 5 pixels to 30 pixels, in concert with the image resolution of 0.1 m. This allowed assessment of the relevance of the approach for practical applications that demand varying degrees of accuracy.

An extracted road border is assumed to be correct if the maximum distance between the extracted road border and its corresponding reference does not exceed the buffer width.

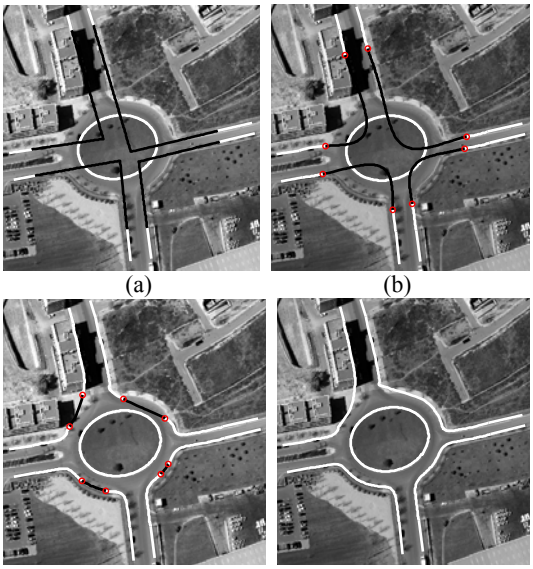


Figure 11. Capture of roundabout outline: (a) initial snakes in black and road arms in white, (b) and (c) evolving curves, and (d) reconstructed roundabout.

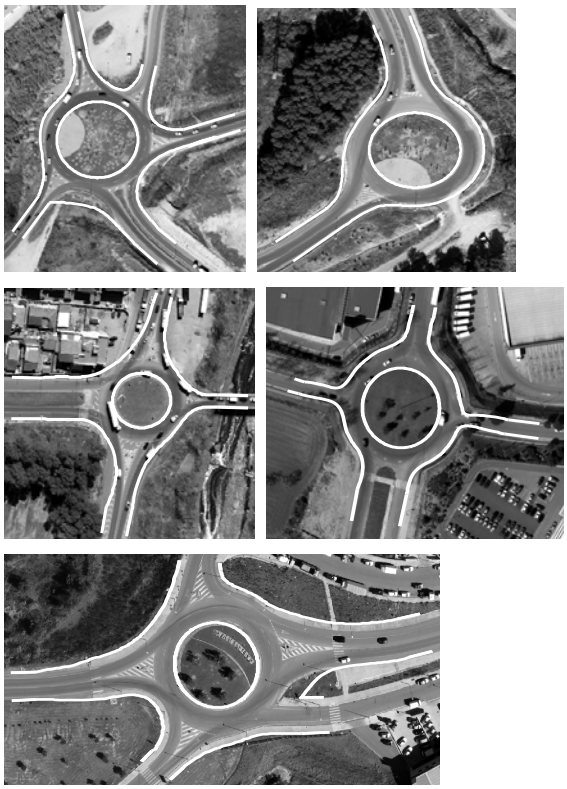


Figure 12. Sample roundabout extraction results for scenes with varying degrees of complexity including disturbances.

A smaller value of the buffer width can be chosen for an application that requires more accurate extraction results. A reference road border is assumed to be matched if the maximum deviation from the extracted object is within the buffer width. Based on these assumptions, three quality measures were adopted, the first being completeness, which is the ratio of the number of matched reference road borders to the number of reference objects. The second is correctness, which is the ratio of the number of correctly extracted road borders to the number of extracted objects, while the third is geometric accuracy, which is expressed by the average distance between the correctly extracted road border and the corresponding reference border, expressed as a Root Mean Square (RMS) value falling within the range of [0, buffer width].

Road border extraction results computed with different buffer width values are shown in Table 1. The completeness of the road border extraction increased as the buffer width value increased from 0.5m to 3m, implying that the results are more complete for higher buffer width values. The geometric accuracy increase is inversely proportional to buffer width so that results obtained with a value of 0.5m are more accurate than those obtained with a larger buffer width. For the buffer width value 0.5 m, the completeness is rather low. The reason is that a slight deviation of the extraction results from the true boundaries exceeding the buffer width frequently occurs due to disturbances and sometimes also due to road markings.

Buffer width (m)	0.5	1	2	3
Number of road borders	41	41	41	41
Completeness	53%	62%	74%	85%
Geometric accuracy (m)	0.30	0.38	0.50	0.58

Table 1. Evaluation results for road borders.

As seen in Table 2, a favourable evaluation result was achieved in the extraction of central islands, which proved the robustness of the proposed method. Central islands of roundabouts were extracted with high values for completeness and correctness for the buffer width of 0.5m, implying the effectiveness of the proposed hybrid evolution strategy. For the buffer width value 1m, all of central islands were extracted correctly.

Buffer width (m)	0.5	1
Number of central islands	10	10
Completeness	90%	100%
Correctness	90%	100%
Geometric accuracy (m)	0.26	0.35

Table 2. Evaluation results for central islands.

5. CONCLUDING REMARKS

A new snake-based approach to automatic extraction of road roundabouts has been described and analysed. Under the approach, the snake's external force field is modified based on the shape of the central island to delineate the roundabout border. The modified snake force field can overcome various disturbances inside and outside the central island. It was shown that the use of prior-knowledge derived from an existing topographic database can considerably enhance the extraction performance. Furthermore, a level set approach with a hybrid evolution strategy was proposed to extract central islands. This produced good results in all 10 test cases, as central islands were extracted correctly for an assigned buffer width of 1m. Nevertheless, partial occlusion of the central island border by large trees and shadowing cannot be overcome at this stage (Fig. 13). There are several possibilities to further enhance the results obtained so far and to be able to deal with more complex scenes. The incorporation of high-level prior knowledge about

the shape of central islands within the level set framework can potentially provide a solution to these problems.

Figure 13. Example of a central island that cannot be extracted due to heavy occlusions caused by trees and shadows.

REFRENCES

Boichis, N., Cocquerez, J.-P., Airault, S., 1998. A top down strategy for simple crossroads extraction. In: *IntArchPhRS.*, Vol. XXXII, Part 2/1, pp. 19-26.

Boichis, N., Viglino, J.-M., Cocquerez, J.-P., 2000. Knowledge based system for the automatic extraction of road intersections from aerial images. In: *IntArchPhRS.*, Vol. XXXIII, Supplement B3, pp. 27-34.

Caselles, V., Catte, F., Coll, T., Dibos, F., 1993. A geometric model for active contours in image processing. *Numer.Math.*, Vol. 66, pp. 1-31.

Cohen, L.D., 1991. On active contours models and balloons. In: *IEEE Transactions on Computer Vision, Graphics, and Image Processing: Image Understanding*, Vol. 53, No. 2, pp. 211-218.

Courant, R., Hilbert, D., 1953. *Methods of Mathematical Physics.* Wiley-Interscience, New York.

De Gunst, M., 1996. Knowledge-based interpretation of aerial images for updating of road maps. Ph.D. thesis, Delft University of Technology, the Netherlands.

Gerke, M., 2006. Automatic Quality Assessment of Road Databases Using Remotely Sensed Imagery. PhD thesis, Leibniz Universität Hannover, Germany, No. 261; also in: Deutsche Geodätische Kommission, Reihe C, No. 599, 105 p.

Heipke C., Mayer H., Wiedemann C., Jamet O., 1998: External evaluation of automatically extracted road axes, *PFG 2*, pp. 81-94.

Kass, M., Witkin, A., Terzopoulos, D., 1988. Snakes: Active contour models. *Int. J. Computer Vision*, 1(4), pp. 321-331.

Li, H., Xu, C., Gui, C., Fox, M.D., 2005. Level Set Evolution Without Re-initialization: A New Variational Formulation. In: *Proceedings of IEEE conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE Computer Society Press., pp. 430-436.

Malladi, R., Sethian, J. A., Vemuri, B. C., 1995. Shape modeling with front propagation: a level set approach. *IEEE Trans. Patt. Anal. Mach. Intell.*, Vol. 17, pp. 158-175.

Neuenschwander, W. M., Fun, P., Iverson, L., Szekely, G., Kubler, O., 1997. Ziplock snakes. *Int. J. Comput. Vis.*, 25(3), pp. 191–201.

Osher, S., Fedkiw, R., 2002. *Level Set Methods and Dynamic Implicit Surfaces*. Sringer-Verlag, New York.

Ravanbakhsh, M., 2008. Road junction extraction from high resolution aerial imagery assisted by topographic database information. PhD thesis, Leibniz Universität Hannover, Germany, No. 273; also in: Deutsche Geodätische Kommission, Reihe C, No. 621, 90 p.

Ravanbakhsh, M., Heipke, C., Pakzad, K., 2008. Road junction extraction from high resolution aerial imagery. *The Photogrammetric Record*, 23(124), pp. 405-423.

U.S. Federal Highway Administration, 2000. Roundabout: An Information guide. FHWA-RD-00-67, http://www.tfhrc.gov/.

Xu. C., Prince, J., 1998. Snakes, shapes, and gradient vector flow. *IEEE Trans. Imag. Proc.*, Vol. 7, pp. 359-369.

ASSESSING THE IMPACT OF DIGITAL SURFACE MODELS ON ROAD EXTRACTION IN SUBURBAN AREAS BY REGION-BASED ROAD SUBGRAPH EXTRACTION

Anne Grote, Franz Rottensteiner

Institute of Photogrammetry and GeoInformation, Leibniz Universität Hannover, 30167 Hannover, Germany
(grote, rottensteiner)@ipi.uni-hannover.de

Commission III, WG III/4

KEY WORDS: High resolution, Aerial, Urban, Automation, Extraction

ABSTRACT:

In this paper, a road extraction approach for suburban areas from high resolution CIR images is presented. The approach is region-based: the image is first segmented using the normalized cuts algorithm, then the initial segments are grouped to form larger segments, and road parts are extracted from these segments. Roads in the image are often covered by several extracted road parts with gaps between them. In order to combine these road parts, neighbouring road parts are connected to a road subgraph if there is evidence that they belong to the same road, such as similar direction and smooth continuation. This process allows several branches in the subgraph which is why another step follows to evaluate the subgraphs and divide them at gaps which show weak connections after gap weights are determined. A digital surface model, if available, is used in the grouping and road extraction step in order to prevent high regions from being extracted as roads. The results of the road extraction with and without the digital surface model are compared in order to show how the extraction is improved by the surface model. It also shows what can still be expected from the extraction if no digital surface model is available.

1. INTRODUCTION

Roads are a very important part of the infrastructure, especially in urban areas. Road data are used in many applications, for example car navigation systems. For these applications it is important that the road data are up-to-date and correct. As the road network is subject to change, especially in suburban areas, the road databases have to be updated frequently. This is often done manually with the help of aerial or satellite images. In order to reduce the costs and the time required for map updating, it is desirable to use automatic procedures for the extraction of roads from these images. Today, roads are to a large degree still extracted manually, especially in urban areas, because of the relatively high complexity of urban environments compared to open landscapes. For open landscapes, road extraction algorithms that are reasonably reliable already exist, e.g. (Zhang, 2004). This was confirmed by the EuroSDR test on road extraction (Mayer et al., 2006). In this test, several state-of-the-art methods for road extraction were compared, using imagery with a resolution of 0.5-1.0 m. The results were reasonably good in rural scenes of medium complexity, but the algorithms did not perform well in urban or suburban areas.

There are many different approaches for road extraction from optical imagery, and in recent years the number of those that deal with urban areas has increased. Road extraction algorithms can be classified into line-based approaches and region-based approaches. Line-based approaches, which model roads as one-dimensional linear objects, are mainly used in open landscapes with images of middle to low resolution, and they are not suitable for urban areas. An approach for urban areas that extracts middle lines and edges of roads and groups them to form road lanes using aerial images of very high resolution (0.1 m) is described by Hinz (2004). In most other approaches regions are extracted from images with a resolution of approximately 1 m. One example is (Zhang and Couloigner, 2006), where a colour image is classified and the regions classified as roads are refined in order to separate roads from false positives such as parking lots. Another example for a region-based approach is (Hu et al., 2007), where footprints of roads are extracted based on shape, and the roads between the footprints are tracked. The high complexity of urban and suburban areas makes road extraction from greyscale aerial images without further information difficult because many different structures in urban areas have an appearance similar to that of roads. Therefore, most approaches use additional information, for example colour (Zhang and Couloigner, 2006; Doucette et al. 2004), Digital Surface Models (DSMs) (Hinz, 2004) or both (Hu et al., 2004). Information about the position of roads from an existing road database can also be used, e.g. (Mena and Malpica, 2005). Prior information about the road network is another possible source of information. Price (1999) assumes that the road network forms a regular grid. This is also done by Youn and Bethel (2004), though they use less strict requirements for the grids.

In this paper, a region-based approach for road extraction from aerial colour images with a resolution of 0.1 m is presented. Optionally, a DSM can be used as an additional source of information. Apart from the DSM, our approach does not require other sources of information such as an existing database, as used in (Mena and Malpica, 2005). Since we work in suburban areas, the approach does not rely on particular properties of roads like road markings, as used in (Hinz, 2004) or a regular road grid, as used in (Price, 1999), and all roads should be extracted, not only major roads. In the approach, an image is first segmented and then road parts are extracted from the segments. These road parts are assembled into road subgraphs. In this way, there is no need to assume that a whole road can be extracted undisturbed. The subgraphs can contain different branches which represent different hypotheses for the

course of the roads. In order to find the most probable course of the road, the subgraphs are evaluated using relations between the road parts and linear programming. If a DSM is available, it can be used in the grouping and road part extraction processes. DSMs have been used in the past for road extraction, but their influence was limited due to the relatively poor performance of standard image matching techniques (Zhang, 2004). We think that with the advent of new dense image matching techniques, e.g. (Pierrot-Deseilligny and Paparoditis, 2006; Hirschmüller, 2008), the importance of incorporating 3D information into the road extraction process will increase. In this paper, the extraction results that were achieved with and without the DSM are compared in order to demonstrate the respective potentials for road extraction. The main goal of this paper is to present the new method for road subgraph evaluation and to assess the influence of the DSM on the road extraction results. The road extraction approach is described in Section 2. The segmentation and road part extraction, which are explained in detail in (Grote et al., 2007; Grote and Heipke, 2008), are only reviewed briefly. Our new method for road subgraph evaluation is discussed in more detail, as well as the incorporation of the DSM. In Section 3, results are presented with a comparison between the results achieved with and without the DSM. Section 4 gives conclusions and directions for future work.

2. APPROACH

2.1 Overview

Our goal is the extraction of roads from high resolution aerial images in suburban areas. We use colour infrared (CIR) images with a ground resolution of approximately 10 cm. Optionally, a DSM, e.g. generated by image matching, can also be used. The approach consists of three steps, namely segmentation, road part extraction and subgraph generation. In the segmentation step, the image is first divided into many small segments, which are then grouped into larger segments having meaningful shapes. Potential road parts are extracted from the grouped segments using shape criteria. If a DSM is available, height can be used as additional criterion in the grouping and road part extraction steps. The road parts are then assembled to road subgraphs (Fig. 1) if they potentially belong to the same road; junctions are not considered in this step. Several branches are allowed to be present in one subgraph. In the next step, these ambiguities are resolved by optimising the graph in a way that finds the best possibility for the course of the road without branches.

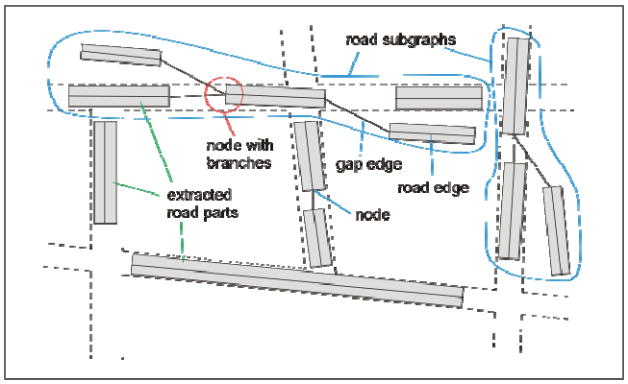


Figure 1. Road subgraphs. Dashed lines: real road network; grey rectangles: extracted road parts; continuous lines: edges of road subgraphs. The blue lines delineate two examples for distinct road subgraphs.

In Fig. 1, the term *road subgraph* and its components are explained. The term *subgraph* is used in order to indicate that it does not represent a complete, interconnected road network. A road subgraph consists of several assembled *road parts*. A road part is a segment which is classified as a road. It can correspond to a whole road between two junctions or only a part of the road, or it can be a false positive. Each subgraph extends only as far as road parts can be found in a more or less straight continuation; in this way, each subgraph usually represents only one road. Each road part in a subgraph has two *nodes* which are connected via a *road edge*. A node can also maintain connections to nodes of other road parts via *gap edges*. These gap edges can be understood as hypotheses for connections between extracted road parts that were missed in the original road part extraction process. If more than one such connection exists at one node, the node has several *branches*. These branches correspond to conflicting hypotheses for a completion of the road. In order to achieve a consistent road network, these conflicts have to be resolved by road subgraph evaluation.

2.2 Segmentation and Road Part Extraction

The first stage of the road extraction is the segmentation of the image, which is carried out in two steps, namely initial segmentation and grouping. The goal of the initial segmentation is to divide the image into small regions whose borders coincide with the road borders as completely as possible. The normalized cuts algorithm (Shi and Malik, 2000) is used for this initial segmentation, in which connections between pixels are weighted according to their similarities. The similarities of pixel pairs are determined using colour and edge criteria. Details can be found in (Grote et al., 2007).

The normalized cuts algorithm results in a considerable oversegmentation. This is necessary in order to preserve most road borders, but as a result, the initial segments must be grouped in order to obtain segments that correspond to road parts. Grouping is carried out iteratively using colour and edge criteria, this time considering the properties of the regions (as opposed to those of the pixels, which were used in the initial segmentation). Segments with irregular shapes that cover roads across junctions can occur in this step. Therefore, the skeletons of the segments are examined. If they have several long branches (not to be confused with the branches of subgraphs), the segments are split.

In the next step, hypotheses for road parts are extracted from the grouped segments. Geometric and radiometric criteria are used for the extraction. The geometric criteria are elongation (ratio of squared perimeter to area), width constancy (ratio of mean width to standard deviation) and difference to average road width. As radiometric criteria, the NDVI (normalized difference vegetation index) and the standard deviation of colour are used. In addition, dark areas are excluded because shadow areas often have similar geometric properties to road parts. The parameters used for the experiments described in this paper are listed in Table 1. The elongation, width constancy, compliance with average road width and the NDVI are used to determine a quality measure for each road part hypothesis. The road parts are represented as regions; for the following road subgraph generation a representation by the centre lines and average widths is also used. For calculating the centre line, the region boundary is split into two parts at the points on the boundary that are farthest away from each other. Distance transforms are calculated for both parts, and the points where

both distance transforms within the road part have the same values make up the centre line. Further details of the road part extraction are explained in (Grote and Heipke, 2008).

min. elongation	70
width	3 m – 16 m
min. width constancy	1.7
min. intensity	40
max. NDVI	0
max std. deviation of colour	50

Table 1. Parameters for road part extraction.

2.3 Road Subgraph Generation and Evaluation

In many cases, a road is not completely covered by one road part but by several different road parts because disturbances in the appearance of the road interfered with the extraction. Therefore, road parts that could belong to the same road are assembled into road subgraphs (Fig. 1) by checking if the road parts have neighbours to which they can be connected. The subgraphs are assembled iteratively, starting with the road part with the best quality measure from the road part extraction. The criteria used to decide whether two road parts belong to the same road are the distance between the segments, the direction difference and the continuation smoothness. The reference points for the measurement of the direction difference and the continuation smoothness are the intersection points between the centre line and the road part borders. The continuation smoothness is measured by calculating the direction differences between the directions of the road parts to the direction of their connection (Fig. 2). The continuation smoothness is high if both smoothness angles are small. However, if the distance between the segments is short, the continuation smoothness criterion is disregarded because at close distances the angles depend too much on the exact positions where the angles are measured.

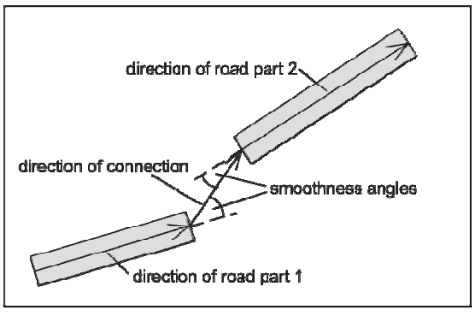


Figure 2. Continuation smoothness.

Two road parts are linked if empirically determined thresholds for the distance, the direction difference and the continuation smoothness are met. The distance and the direction difference must be low and the continuation smoothness must be high; all three conditions must be fulfilled for the road parts to be linked. The parameters used for the experiments described in this paper are shown in Table 2. One road part can be attached to more than one other road part in the same direction, such that branches in the subgraphs can occur. The search for neighbouring road parts continues until no more road parts can be added. Then, the search is resumed with the road part which has the best quality measure among the remaining road parts until all road parts have been examined.

max. distance	50 m
max. direction difference	40°
max. smoothness angle	40°

Table 2. Parameters for road subgraph generation.

In most cases the branches do not represent actual branches in the road network but rather indicate false extractions of road parts that are nearly parallel to the real road. Therefore, road subgraphs containing branches are treated as including different hypotheses for the course of the road. It is the goal of road subgraph evaluation to determine the best hypotheses, i.e. the hypotheses that are most likely to actually correspond to roads, and to discard all the other hypotheses. This goal is achieved via the formulation and solution of a linear programming problem.

In linear programming a linear function (objective function) whose variables are subject to linear constraints is maximised or minimised (Dantzig, 1963). The constraints define a set of feasible vectors; the vector for which the constraint set is maximal or minimal is the optimal solution for the problem. Linear programming can be used when the variables of the linear function to be optimised are restricted by hard constraints, which can be described by equations or by inequalities. In our case the constraints are inequalities resulting from the condition that no node of a subgraph should be connected to more than one gap edge after the optimisation. The objective function which is to be maximised is

$$w_1 x_1 + \ldots + w_n x_n \rightarrow \max \qquad (1)$$

where $w_1 \ldots w_n$ are weights of the gap edges that reflect the plausibility that the two road parts belong to the same road. The unknown variables are $x_1 \ldots x_n$. There is one such unknown for each gap edge in the road subgraph. Each of these binary variables indicates whether its corresponding edge should be kept or discarded. A value of 1 indicates that the edge is kept; a value of 0 indicates that it is discarded. These values are determined by solving the maximisation under the constraints that each node i can only be associated to one gap edge:

$$\sum_{j \in E_i} x_j \leq 1 . \qquad (2)$$

E_i is the set of gap edges belonging to node i. The optimisation is carried out using the simplex method (Dantzig, 1963). The edge weights are determined using the following criteria:

- Distance: a shorter distance between the two connected road parts gives a higher edge weight.
- Road part quality: the sum of the quality measures of both road parts from the extraction. A higher value gives a higher edge weight.
- Colour: a smaller difference between the mean colour values of both road parts gives a higher edge weight.
- Width: a smaller width difference between both road parts gives a higher edge weight.
- Continuation smoothness: smaller smoothness angles (cf. Section 2.2) give a higher edge weight.
- Direction: a smaller direction difference between both road parts gives a higher edge weight.

The weights for the different criteria are obtained after calculating all criteria by mapping the respective values linearly onto an interval between 0 and 1. For example, the maximum possible distance between two connected road parts is equivalent to a distance weight of 0. The other weights are obtained accordingly. All weights are multiplied to obtain the total weight for one edge. The edge weights that belong to the same subgraph are normalised such that their sum equals 1.

After solving the linear programming problem, gap edges whose corresponding unknowns were determined to be 0 are discarded. This results in consistent road subgraphs that correspond to roads, which are considered to be the results of road extraction. However, all road parts are kept at this stage, so the falsely extracted road parts must be removed during the road network formation, which is still under development.

2.4 Including the Digital Surface Model

The road part extraction can produce false positives among segments having properties that are similar to road segments. False positives can disturb the later steps of linking the road parts and forming a road network. To avoid this, they have to be sorted out later when the network is formed or they have to be prevented from being extracted. The majority of the falsely extracted road parts are buildings, which can lead to subgraphs consisting only of false positives because buildings are often arranged in rows. As the most distinctive property of buildings that distinguishes them from roads is their height, a DSM can provide valuable additional information.

The DSM is employed in the grouping and road part extraction steps. It is not used for the initial segmentation which operates at pixel level because DSM inaccuracies in shadows and alignment errors caused by the fact that orthophotos are generated using a Digital Terrain Model (DTM) would affect the results adversely. In the grouping step the DSM is used to prevent segments with different heights from being merged. For this purpose, the differences of the mean heights are added to the grouping criteria. If the difference is larger than a threshold, the segments are not merged. The threshold is empirically determined; in our examples it is set to 1.5 m. This prevents building segments from being merged with road segments but allows for smaller height variations in the background.

For the road part extraction the DSM is used to prevent high objects from being extracted as roads. For this purpose, a normalised DSM (nDSM) representing objects above ground is determined. A coarse Digital Terrain Model (DTM) is generated from the DSM by morphological grey value opening. The DTM is then subtracted from the DSM, which yields the nDSM (Weidner and Förstner, 1995). The mean heights of the segments obtained from the nDSM are compared to a threshold. It was found that a threshold of about 1 m reliably distinguishes building parts and road parts. This threshold is used as additional criterion in the road part extraction.

3. RESULTS

The approach was tested on three subsets of an image showing a suburban scene from Grangemouth, Scotland. The image is a CIR orthoimage with a resolution of 10 cm. The data set also contains a DSM that was generated by image matching at a resolution of 20 cm in position and 10 cm in elevation. Elevated objects are represented well in the DSM, though unfortunately it is not known which method was used for its generation. For the three subsets, results of the road part extraction and road subgraph generation are presented, first obtained from the image data alone, and then from additionally using the DSM.

3.1 Results without DSM

Segmentation, grouping and road part extraction were carried out as described in Section 2.2 and (Grote et al., 2007; Grote and Heipke, 2008). Figures 3, 4 and 5 show the results of the road part extraction for the image subsets 1, 2 and 3, respectively. Whereas in subsets 1 and 2 most parts of the road network were extracted, significant parts of the road network are missed in subset 3. Each subset contains false positives, which are mainly found on buildings because buildings have similar radiometric and geometric properties to road parts. The results of the road part extraction were compared to manually extracted road regions. The manually extracted regions include areas occluded by shadows or trees, but exclude pavements. The completeness and correctness of the road parts computed according to (Heipke et al., 1997) are displayed in Table 3. They were determined on a per-pixel level and thus refer to the extracted areas. Table 3 shows that about two thirds of the road area could be detected, but almost half of the area classified as road area consists of false positives.

	Completeness	Correctness
Subset 1	66 %	57 %
Subset 2	89 %	59 %
Subset 3	31 %	49 %
Total	62 %	55 %

Table 3. Evaluation of road part extraction without a DSM.

Figure 3. Road parts extracted in subset 1 (yellow).

Figure 4. Road parts extracted in subset 2 (yellow).

Figure 5. Road parts extracted in subset 3 (yellow).

The road parts are then assembled into road subgraphs as is shown in Fig. 6 for subset 1. There are three subgraphs which contain different hypotheses; these are resolved using linear programming, as described in Section 2.3. In Fig. 7, only these three subgraphs are shown with the edges that are removed displayed in red. The results show that the optimisation favours connections between road parts that are similar in colour and width and maintain a more or less straight continuation.

Figure 6. Road subgraphs (without DSM) for subset 1. Different colours represent different road subgraphs.

Figure 7. Road subgraph evaluation (without DSM) for subset 1. Discarded gap edges are displayed in red.

3.2 Results with DSM

The grouping and the road part extraction were repeated using the DSM as additional information, as described in Section 2.4. Figures 8, 9 and 10 show the results of the road part extraction with the DSM for the image subsets 1, 2 and 3, respectively. Both the completeness and the correctness values (Table 4) have notably improved compared to the results without the DSM. The highest improvement in completeness is observed in subset 3; almost all roads are now covered with road parts for the greater part of their area. The highest improvement in correctness is observed in subset 1 where several buildings were extracted without the DSM.

	Completeness	Correctness
Subset 1	73 %	73 %
Subset 2	91 %	65 %
Subset 3	45 %	57 %
Total	70 %	65 %

Table 4. Evaluation of road part extraction with a DSM.

The subgraph generation and evaluation is conducted in the same way as before. The subgraphs for subset 1 can be seen in Fig. 11. Now there is only one subgraph with several branches, because the use of the DSM prevented some buildings from being extracted. The result of the evaluation of the remaining subgraph with branches is shown in Fig. 12.

Figure 8. Road parts extracted in subset 1 with DSM (yellow).

Figure 9. Road parts extracted in subset 2 with DSM (yellow).

Figure 10. Road parts extracted in subset 3 with DSM (yellow).

Compared to the visual impression of the extracted roads, the completeness and correctness values are relatively low. The computed correctness suffers from leakage at the borders of the road parts and from the fact that pavements, which are often extracted as roads, are not included in the reference data. The computed completeness would also be increased by constructing road parts corresponding to the gap edges that were accepted in subgraph evaluation.

Figure 11. Road subgraphs (with DSM), subset 1. Different colours represent different road subgraphs.

Figure 12. Road subgraph evaluation (with DSM), subset 1.

4. CONCLUSIONS

In this paper, an approach for the extraction of roads in suburban areas was presented, with the focus on a comparison between the extraction results achieved for image data alone and the results achieved for using a DSM as an additional information source. Our results show that the approach is suitable for the extraction of roads in suburban areas. The majority of roads can be detected even without a DSM, though there is a relatively high number of false positives, mostly buildings. Using a DSM improves both the completeness and the correctness of the results, primarily because buildings can now be clearly separated from roads. The correctness is improved because buildings are not extracted as false positives. The completeness is improved because incorporating the DSM into the grouping process provides a better grouping result from which more road parts can be extracted. Without a DSM, there are more subgraphs containing several branches, so that the importance of the subgraph evaluation is higher. The potential to find the real course of the road based on an optimisation of the interrelations between the road parts is shown in Figures 6 and 7. Subgraph evaluation can thus compensate for the lack of height information in the road part extraction stage. However, the improvement caused by using the height information in the grouping phase cannot be compensated. Road parts that remain undetected due to a poor performance of grouping based on image data alone cannot be detected at a later stage. Using a DSM thus certainly improves the quality of the results. This can be seen in particular for subset 3 (Fig. 5 vs. Fig. 10).

The road extraction process can still be improved in several ways. The parameters used for grouping, road part extraction and road subgraph generation could be learned from training samples, which probably would improve stability in different settings. The road extraction can also be improved by incorporating context objects such as trees, buildings and vehicles. It is planned to incorporate context objects into the evaluation of the gaps within the subgraphs, combined with the

interrelations described in this paper. Context objects can also be beneficial in the next steps, which include the formation of a road network by searching for junction hypotheses between road strings and removing isolated (mainly falsely extracted) road parts. The completeness and correctness values given in this paper where obtained from the single road parts. They are likely to improve during the network generation because the majority of falsely extracted road parts can be removed and the gaps between road parts in a string can also be counted as road.

ACKNOWLEDGEMENTS

This project is funded by the DFG (German Research Foundation) under grant HE 1822/21-1. The calculation of the normalized cuts was made with a C++ program partly adapted from a MATLAB program written by Timothée Cour, Stella Yu and Jianbo Shi. Their program can be found at www.seas.upenn.edu/~timothee/software_ncut/software.html (accessed March 2009). The calculation of the linear program was made with the MILP solver lp_solve (lpsolve.sourceforge.net/5.5/, accessed March 2009).

REFERENCES

Dantzig, G.B., 1963. Linear programming and extensions. Princeton University Press, Princeton, New Jersey, USA.

Doucette, P., Agouris, P. and Stefanidis, A., 2004. Automated road extraction from high resolution multispectral imagery. *PE & RS* 70(12), pp. 1405-1416.

Grote, A., Butenuth, M. and Heipke, C., 2007. Road extraction in suburban areas based on normalized cuts. In: *IAPRSIS* XXXVI-3/W49A, pp. 51-56.

Grote, A. and Heipke, C., 2008. Road extraction for the update of road databases in suburban areas. In: *IAPRSIS* XXXVII-B3b, pp. 563-568.

Heipke, C., Mayer, H., Wiedemann, C. and Jamet, O., 1997. Evaluation of automatic road extraction. In: *IAPRS* XXXII-3/2W3, pp. 47-56.

Hinz, S., 2004. Automatic road extraction in urban scenes – and beyond. In: *IAPRSIS* XXXV-B3, pp. 349-355.

Hirschmüller, H., 2008. Stereo processing by semi-global matching and mutual information. *IEEE TPAMI* 30(2), pp. 328-341.

Hu, J., Razdan, A., Femiani, J.C., Cui, M. and Wonka, P., 2007. Road network extraction and intersection detection from aerial images by tracking road footprints. *IEEE TGRS* 45(12), pp. 4144-4157.

Hu, X., Tao, C.V. and Hu, Y., 2004. Automatic road extraction from dense urban area by integrated processing of high resolution imagery and LIDAR data. In: *IAPRSIS* XXXV-B3, pp. 288-292.

Mayer, H., Hinz, S., Bacher, U., and Baltsavias, E., 2006. A test of automatic road extraction approaches. In: *IAPRSIS* XXXVI-3, pp. 209-214.

Mena, J.B. and Malpica, J.A., 2005. An automatic method for road extraction in rural and semi-urban areas starting from high-resolution satellite imagery. *Pattern Recognition Letters* 26(9), pp. 1201-1220.

Pierrot-Deseilligny, M. and Paparoditis, N., 2006. A multiresolution and optimization-based image matching approach: an application to surface reconstruction from SPOT-HRS stereo imagery. In: *IAPRSIS* XXXVI-1/W41, pp. 73-77.

Price, K., 1999. Road grid extraction and verification. In: *IAPRS* XXXII- 3/2W5, pp. 101-106.

Shi, J. and Malik, J., 2000. Normalized cuts and image segmentation. *IEEE TPAMI*. 22(8), pp. 888-905.

Weidner, U. and Förstner, W. 1995. Towards automatic building reconstruction from high resolution digital elevation models. *ISPRS J. Photogr. & Rem. Sens.* 50(4), pp. 38–49.

Youn, J. and Bethel, J.S., 2004. Adaptive snakes for urban road extraction. In: *IAPRSIS* XXXV-B3, pp. 465-470.

Zhang, C., 2004. Towards an operational system for automated updating of road databases by integration of imagery and geodata. *ISPRS J. Photogr. & Rem. Sens.* 58(3-4), pp. 166-186.

Zhang, Q. and Couloigner, I., 2006. Automated road network extraction from high resolution multi-spectral imagery. In: *Proc. ASPRS Annual Conf.*, Reno, Nevada, 10 p., on CD-ROM.

VEHICLE ACTIVITY INDICATION FROM AIRBORNE LIDAR DATA OF URBAN AREAS BY BINARY SHAPE CLASSIFICATION OF POINT SETS

W. Yao [a, *], S. Hinz[b], U. Stilla [a]
[a]Photogrammetry and Remote Sensing, Technische Universitaet Muenchen, Arcisstr.21, 80290 Munich, Germany
[b]Institute of Photogrammetry and Remote Sensing, Universität Karlsruhe (TH), 76128 Karlsruhe, Germany

KEY WORDS: Airborne LiDAR, Urban areas, Vehicle extraction, Motion indication, Shape analysis

ABSTRACT:
This paper presents a generic scheme to analyze urban traffic via vehicle motion indication from airborne laser scanning (ALS) data. The scheme comprises two main steps performed progressively — vehicle extraction and motion status classification. The step for vehicle extraction is intended to detect and delineate single vehicle instances as accurate and complete as possible, while the step for motion status classification takes advantage of shape artefacts defined for moving vehicle model, to classify the extracted vehicle point sets based on parameterized boundary features, which are sufficiently good to describe the vehicle shape. To accomplish the tasks, a hybrid strategy integrating context-guided method with 3-d segmentation based approach is applied for vehicle extraction. Then, a binary classification method using Lie group based distance is adopted to determine the vehicle motion status. However, the vehicle velocity cannot be derived at this stage due to unknown true size of vehicle. We illustrate the vehicle motion indication scheme by two examples of real data and summarize the performance by accessing the results with respect to reference data manually acquired, through which the feasibility and high potential of airborne LiDAR for urban traffic analysis are verified.

1. INTRODUCTION

Transportation represents a major segment of the economic activities of modern societies and has been keeping increase worldwide which leads to adverse impact on our environment and society, so that the increase of transport safety and efficiency, as well as the reduction of air and noise pollution are the main task to solve in the future (Rosenbaum et al., 2008). The automatic extraction, characterization and monitoring of traffic using remote sensing platforms is an emerging field of research. Approaches for vehicle detection and monitoring rely not only on airborne video but on nearly the whole range of available sensors; for instance, optical aerial and satellite sensors, infrared cameras, SAR systems and airborne LiDAR (Hinz et al., 2008). The principal argument for the utilization of such sensors is that they complement stationary data collectors such as induction loops and video cameras mounted on bridges or traffic lights, in the sense that they deliver not only local data but also observe the traffic situation over a larger region of the road network. Finally, the measurements derived from the various sensors could be fused through the assimilation of traffic flow models. The broad variety of approaches can be found, for instance, in compilations by Stilla et al., (2005) and Hinz et al., (2006).

Nowadays, airborne optical cameras are widely in use for these tasks(Reinartz et al., 2006). Yet satellite sensors have also entered into the resolution range (0.5-2m) required for vehicle extraction. Sub-metric resolution is even available for SAR data since the successful launch of TerraSAR-X. The big advantage of these sensors is the spatial coverage. Thanks to their relatively short acquisition time and long revisit period, satellite systems can mainly contribute to the collection of statistical traffic data for validating specific traffic models. Typical approaches for vehicle detection in optical satellite images are described by Jin and Davis, (2007) and Sharma et al., (2006), and in spaceborne SAR images by Meyer et al., (2006) and Runge et al., (2007). For monitoring major public events, mobile and flexible systems which are able to gather data about traffic density and average speed are desirable. Systems based on medium or large format cameras mounted on airborne platforms meet the demands of flexibility and mobility. With them, large areas can be covered (up to several km^2 per frame) while keeping the spatial resolution high enough to image sufficient detail. A variety of approaches for automatic tracking and velocity calculation from airborne cameras have been developed over the last few decades. These approaches make use of substructures of vehicles such as the roof and windscreen, for matching a wire-frame model to the image data (Zhao and Nevatia, 2003).

Despite that LiDAR has a clear edge over optical imagery in terms of operational conditions, there have been so far few works conducted in relation to traffic analysis from laser scanners. On the one hand it is an active sensor that can work day and night; on the other hand it is range senor that can capture 3d explicit description of scene and penetrate volumetric occlusions to some extent. Toth and Grejner-Brzezinska, (2006) has presented an integrated airborne system of digital camera and LiDAR for road corridor mapping and dynamical information acquisition. They addressed a comprehensive working chain for near real-time extracting vehicles motion based on fusing the images with LiDAR data. Another example of applying ALS data for traffic-related analysis can be found in Yarlagadda et al., (2008), where the vehicle category is determined by 3-d shape-based classification.

In this paper, a generic scheme to discover the vehicle motion solely from airborne LiDAR data is presented. It is based on two-step strategy, which firstly extracts single vehicles with contextual model of traffic objects and 3d-segmentation based classification (3-d object-based classification), and secondly classifies vehicle entities in view of motion status based on shape analysis.

2. VEHICLE EXTRACTION

In this step, we need to at first extract various vehicle categories as complete and accurate as possible, but not considering the difference among them in terms of dynamical status. To

* Corresponding author.

accomplish this task, we proposed a hybrid strategy that integrates context-guided progressive method with 3-d segmentation based classification. Experiments demonstrated that the assimilation of these two approaches (Fig. 1) can make our vehicle extraction from LiDAR data of urban areas more competent and robust, even against complex scenes.

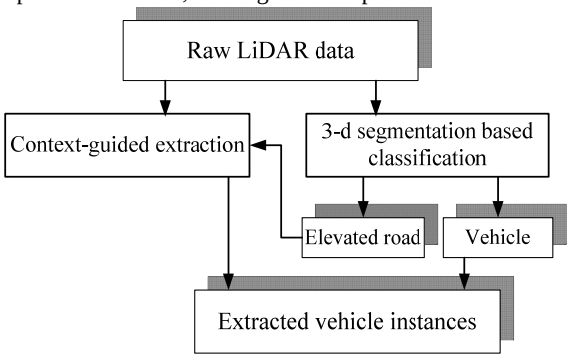


Figure 1. Integrated scheme for vehicle extraction.

2.1 Context-guided extraction

This extraction strategy comprises knowledge about how and when certain parts of the vehicle and context model of traffic related objects in urban areas are optimally exploited, thereby forming the basic control mechanism of the extraction process. In contrast to other common approaches dealing with LiDAR data analysis, it neither uses the reflected intensity for extraction nor combines multiple data sources acquired simultaneously. The philosophy is to exploit geometric information of ALS data as much as possible primarily based on such context-relation that vehicles are generally placed upon the ground surface. Moreover, the approach on the one side can be viewed as a processing strategy progressively reducing "region of interest". It is subdivided into four steps: ground level separation, geo-tiling and filling, vehicle-top detection and selection, segmentation, which are respectively elaborated in detail in Yao et al., (2008)a. An exemplary result on one co-registered dataset is shown in Fig.2.

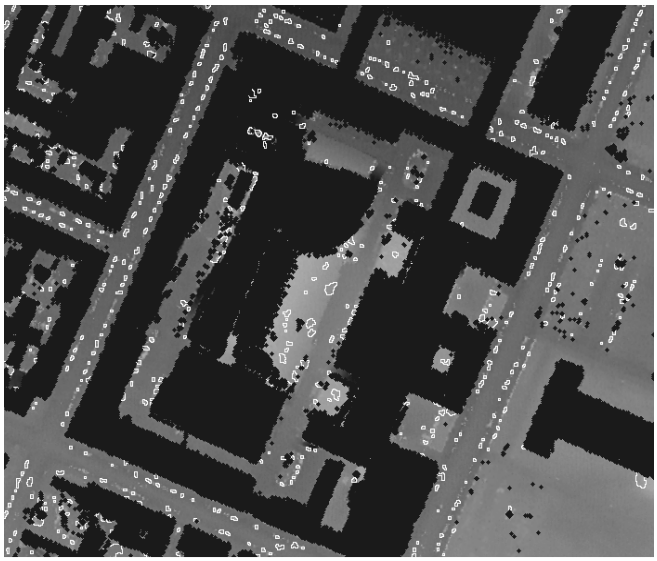


Figure 2. Vehicle extraction result as white outlined contours for test data I using context-guilded method.

2.2 3D segmentation based classification

Since many vehicles in modern cities might travel on the elevated roads such as flyover or bridge, the context relation abided by the method in section 2.1 does not always hold. Therefore, we introduced a 3D object-based classification strategy for extracting semantic objects directly from LiDAR point cloud of urban areas. It could either extract two object classes – vehicle and elevated road simultaneously or only elevated road, where vehicle can further be detected considering elevated road here as ground. The ALS data is firstly subjected to the segmentation process using nonparametric clustering tool – mean shift (MS). The obtained results are usually not able to give a significative description of distinct natural and man-made objects in complex scenes, even though MS does a genuine clustering directly on 3D point cloud to discover various geometric modes in it. Hence, the initial resulted point segments have to be handled under the global optimization criterions to generate more consistent subsets of laser data. For it, a modified normalized-cuts (Ncuts) is applied with the sense of perceptual grouping. Finally, based on derived features of spatially separated point clusters that potentially correspond to semantic object entities, classification is performed to evaluate them to extract the flyover and vehicle (Yao et al., 2009). Applying this approach to a one-path dataset yielded Fig.3.

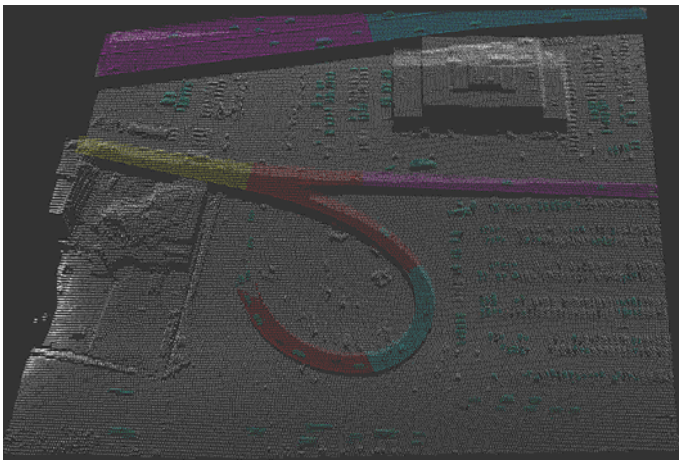


Figure 3. Vehicle (green) and flyover extraction results for test data II using 3D segentation based classification.

3. VEHICLE MOTION INDICATION

For extracted vehicles resulted from last step, the parameterized model for point sets of single vehicles can then be produced by shape analysis. From the parameterized features of vehicle shape, the across-track vehicle motion (-component) is able to be indicated unambiguously based on the moving vehicle model in ALS data, whereas the along track motion cannot be implied without prior knowledge about individual vehicle sizes. In this section, the vehicle motion status is attempted to be inferred up to the across-track direction without derive the velocity.

3.1 Vehicle Parametrization

Generally, the laser data provide us a straightforward 3D parameterization, as vehicle forms change more vertically than horizontally. To refine the 3D vehicle envelope model (Yao et al., 2008b), however, is difficult, because the laser point density acquired under common configurations is usually not sufficient to model the vertical profile of a vehicle. The situation is even more degraded by motion artifacts, because the large relative velocity of the sensor to object results in fewer laser points, making vehicle appears like a blob. Consequently, it is not easy to analytically model the vertical vehicle profiles from ALS data, which would be a simple task for much denser terrestrial laser data.

Yarlagadda et al., (2008) has applied a spoke model to vehicle database in a parking lot scanned by airborne LiDAR for 3D classification task of vehicle category. The point cloud of single vehicle is fitted with multiple connected planes being similar to spokes, which are used to describe the vehicle shape via two controlling parameters for each spoke, namely the orientation and radius of it. For the purpose of our task, it is desirable that the original vehicle form and motion artifacts are able to be captured by a unified geometric model. Due to flexibility and efficiency, the spoke model for vehicle point sets is selected here as general framework for vehicle shape parametrization. Being subject to minor modifications towards the analysis objective, the spoke model could consistently encode geometric information used for robust classification of vehicle motion.

Based on the moving vehicle model, which is focused on the 2-d deformation of vehicle form, the 3D spoke model of vehicles can be projected onto 2-d plane to deriving the shape parameters, thereby avoiding unnecessary complexities. Instead, the angle of shear and radius of projected 2-d point sets have to be estimated as controlling parameters of modified spoke model for vehicle parametrization. Due to the limited point sampling rate of ALS data, the number of spokes in the model is flexible to be determined depending on the point density or vehicle category, despite that the vehicles in our test data are frequently modeled with only one spoke.

To obtain the geometric features of extracted vehicles, the shape analysis is to be performed on the projected point sets of the spoke model. The whole procedure mainly consists of two steps: boundary tracking and parallelogram fitting.

A modified convex-hull algorithm (Jarvis, 1977) is used to determine the boundary of a set of points, namely the spoke model of extracted vehicles. The modification is to constrain the searching space of a convex hull formation to a neighborhood. The study showed that the approach can yield satisfactory results if the point distribution is consistent throughout the dataset. Such condition could be fulfilled, as only one-path ALS data are considered for moving object. The boundary tracing method for a point set B using a modified convex hull analysis starts also with a randomly selected boundary point P. Then, we use the convex hull algorithm to find the next boundary point P_k within the neighborhood of P, which is defined as rectangle with two dimensions corresponding to the point spacing in along and across-track directions of ALS data. Finally, the approach will proceed to the newly selected boundary point and repeat the step mentioned above until the point P is selected as P_k again, as depicted in the left column of Fig.4.

Since the sampling irregularity and randomness are generally assumed to be present in the LiDAR data, the traced boundary cannot be directly used as shape description for single vehicle instances, based on which the shape analysis is performed to parameterize the vehicle point sets. Consequently, a boundary regularization process aided by analytic fitting operations is to be introduced for tackling these problems. It is noticed that most vehicles have mutually parallel directions. We can find these directions from the boundary points and fit parametric lines.
The first step in regularization is to extract the points that lie on identical line segments. This is done by sequentially following the boundary points and locating positions where the slopes of two consecutive edges are significantly different. Points on these edges are grouped to one line segment. Therefore, a set of line segments $\{l_1, l_2, ..., l_n, n \geq 4\}$ from which four longest line segment $\{L_1, L_2, L_3, L_4\}$ are selected. Each of the selected line segments is modeled by equation $A_i x + B_i y + 1 = 0$. Based on the slope $M_i = -A_i / B_i$, line segments are sorted into different groups, each of which contains line segments being parallel within a given tolerance. As we know from the defined vehicle models (Yao et al., 2008b), the vehicle point sets generally appear as a parallelogram and have only two groups of line segments, i.e. vertical and horizontal.

The next step is to determine the least squares fitting to these line segments, with the constraints that the lines segments are parallel to each other within one group, namely parallelogram fitting. The solution consists of sets of parameters required to describe four line segments, which are formed as following line equations:

$$A_i x + B_i y + 1 = 0 \qquad i = 1,2,3,4;\ j = j(\mathrm{i}) = 1,2,3,\ldots m_i$$

with the condition: $\begin{matrix} M_1 = M_3 \\ M_2 = M_4 \end{matrix} \Leftrightarrow L_1\ (L_2)$ and $L_3\ (L_4)$ are opposite sides.

where m_i is the number of points on the line segment i. However, there are no specific constraints on the line segments belonging to different groups.

Once the spoke model of vehicle point sets is constructed and parameterized (Fig.4, right column), two controlling parameters can be derived, which measure the accordance of 2-d point sets to parallelogram (non-rectangularity) and dimension scale, respectively. The angle of shear θ_{SA} of parameterized vehicle point set:

$$\theta_{SA} = \arctan\left(\left|\frac{M_2 - M_1}{1 + M_1 \cdot M_2}\right|\right),$$

The extent E of parameterized vehicle point set:

$$E = |L_1| \cdot |L_2| \cdot \sin\theta_{SA}$$

where M_2, M_1 are slopes of line segments belonging to two groups respectively and $|\ |$ indicates the length of corresponding line segment.

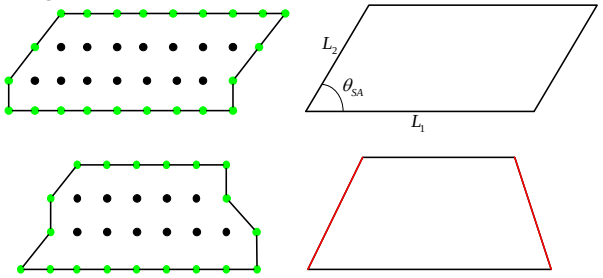


Figure 4. Two examples for vehicle parametrization: boundary tracing, shape regulation (parallelogram fitting). Top row: moving vehicle; bottom row: vehicle of ambiguous movement with abnormal laser reflections. Green points marks the borders of extracted vehicle, red lines indicate the non-parallel sides of a fitted vehicle shape.

Two basic cases have to be distinguished in view of vehicle movement, based on the geometric features derived above for each extracted vehicle. However, they occasionally emerge

other than as parallelogram (Fig.4, bottom row), but e.g. trapezoid, common quadrilaterals, etc, due to unstable sampling characteristics of LiDAR or clutter objects in urban areas. It is difficult to decide whether it is actually a moving vehicle part or a point set of stationary vehicle with missing parts. Generally, these vehicle point sets confuse the shape analysis and deliverer only ambiguous geometric features that cannot be adopted for robust classification. Therefore, this category of vehicle point sets have to be identified and then excluded from candidates delivered to movement classification, which means that they could be only attributed to uncertain motion status at the moment. Those point sets are also undergone the same shape analysis process and can be found when the parallelogram fitting fails.

3.2 Movement classification

As indicated in section 3.1, the point sets of extracted vehicle can generally be denoted by spoke model with two parameters, whose configuration is formulated as

$$X = \begin{pmatrix} U_1 \\ . \\ . \\ U_k \end{pmatrix}, \quad U_i = \begin{pmatrix} \theta_{SA}^i \\ E_i \end{pmatrix}$$

where k denotes the number of spokes in the model. As inspired by the works of Fletcher et al., (2003) and Yarlagadda et al., (2008), the 3D vehicle shape variability is nonlinear and represented as a transformation space. Thus the similarity between vehicle instances can be measured by group distance metric. It has been also confirmed that Lie group PCA can better describe the variability of data that is inherently nonlinear and statistics on linear models may benefit from the addition of nonlinear information. Since our task is intended to classify the vehicle motion based on the shape features of vehicle point sets, the classification framework for distinguishing generic vehicle category can be easily adapted to motion analysis.

Consequently, a new vehicle configuration Y can be obtained by a transformation of X written in matrix form: $Y=T(X)$ where

$$T = \begin{pmatrix} M_1 & . & 0 \\ . & . & . \\ 0 & . & M_k \end{pmatrix}, \quad M_i = \begin{pmatrix} R_i & 0 \\ 0 & e^{ai} \end{pmatrix}$$

, R_i denotes the 2-d rotation acting on the angle of shear θ_{SA} . e^{ai} denotes the scale acting on the extent E. By varying T, different vehicle shape (motion status) can be represented as transformations of X. based on elaborations in Rossmann (2002), M_i is a Cartesian product of the scale and angle value groups $\Re \times \mathbf{SO}(2)$, which are the Lie group of 1-d real value and the Lie group of 2-d rotation, respectively. Since the Cartesian product of Lie group elements is a Lie group and T is the Cartesian product of transformation matrices M acting on the individual spokes, T forms a Lie group. The group T is a transformation group that acts on shape parameters M. However, any vehicle shape X may be represented in T as the transformation of a fixed identity atom.

A group is defined as a set of elements together with a binary operation (multiplication) satisfying the closure, associative, identity and the inverse axioms. A Lie group G is a group defined on differentiable manifold. The tangent space of group G at the identity e, T_e , is called the Lie algebra g. The exponential map *exp* is a mapping from Lie algebra elements to Lie group elements. The inverse of the exponential map is called logarithmic map *log*. The Lie algebra element of T is obtained by performing component-wise log operation on each of the M_i :

$$\log(T) = \begin{pmatrix} \log(M_1) & . & 0 \\ . & . & . \\ 0 & . & \log(M_k) \end{pmatrix} \qquad (1)$$

where $\log(M_i) = \alpha_i \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} + \theta_i \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}$. Equation (1) expresses the Lie algebra element of an individual spoke in terms of the generator matrices for scaling and 2-d rotation factors.

The intrinsic mean μ of a set of transformation matrices T_1 , T_2 , …, T_n of vehicle spoke models is defined as

$$\mu = \arg\min \sum_{k=1}^{n} d(T_1, T_2)^2 \qquad (2)$$

where $d(\cdot,\cdot)$ denotes Riemannian distance on G, and $d(T_1,T_2) = \left\| \log(T_1^{-1}T_2) \right\|$ where $\|\cdot\|$ is the Frobenius norm of the resulting algebra elements. The 1-parameter Lie algebra element of the spoke model of vehicle point sets is given by

$$A_v(t) = \begin{pmatrix} A_{v_1}(t) & . & 0 \\ . & . & . \\ 0 & . & A_{v_n}(t) \end{pmatrix} \qquad (3)$$

where $A_{v_i}(t) = t\log(M_i)$, denoting that the Lie algebra element is defined at a fixed (α_i, θ_i) for each spoke, which represents the tangent to a geodesic curve parameterized by t. The parameter t in (3) sweeps out a 1-parameter sub-group, $H_v(t)$ of the Lie group G of spoke transformations. For any $g \in G$, the distance between g and $H_v(t)$ is defined as

$$d(g, H_v) = \min d(g, \exp[A_v(t)]), \quad t \in \Re \qquad (4)$$

Analogous to the principle components of a vector space, there exist 1-parameter subgroups called the principle geodesic curves (Fletcher et al., 2003) which describe the essential variability of the data points lying on the manifold. The first principle geodesic curve for elements of a Lie group G is defined as the 1-parameter subgroup $H_{v^{(1)}}(t)$, where

$$v^{(1)} = \arg\min \sum_{i=1}^{n} d^2(\mu^{-1}g_i, H_v) \qquad (5)$$

Let $p_{i,1}$ be the projection of $\mu^{-1}g_i$ on $H_{v^{(1)}}$, and define $g_i^{(1)} = p_{i,1}^{-1}\mu^{-1}g_i$. The higher k-th principle geodesic curve can be determined recursively based on (5).

The motion analysis can then be formulated as a binary classification problem using Lie distance metrics. The input to the Lie distance classifier comprises a set of labeled samples T_j from two categories of vehicle status C_j - moving vehicles and stationary ones. n_Y denotes the number of training samples for each category. The intrinsic mean μ_j and the principal geodesics $H_{v^{(n)}}$ are computed for each vehicle class C_j using

the samples $S_j^m \in S_j$, $1 \le m \le n_Y$. Once the principal geodesics are available for each C_j, the classification of an unlabeled sample x can be performed by finding the category with the closest first principal geodesics to x. The corresponding motion status of a vehicle is found by

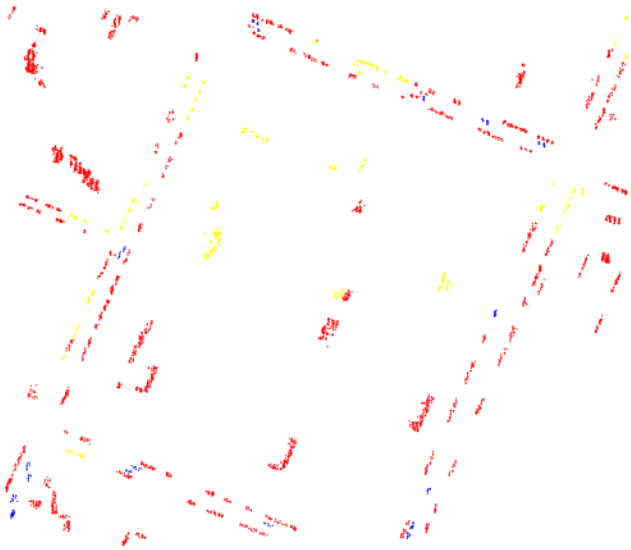
$$j^* = \arg\min \left\| \log(H_{j,v^{(1)}}^{-1} x) \right\|, \quad j \in \{1,2\} \qquad (6)$$

Generally, it is claimed that the classification of vehicle status can successfully run based solely on the first principal geodesics of a movement category. Although there are significant variations in shape over one category, the first principal geodesics $H_{v^{(1)}}$ is assumed to summarize the essential shape features of vehicle point sets in terms of only distinguishing between binary motion statuses.

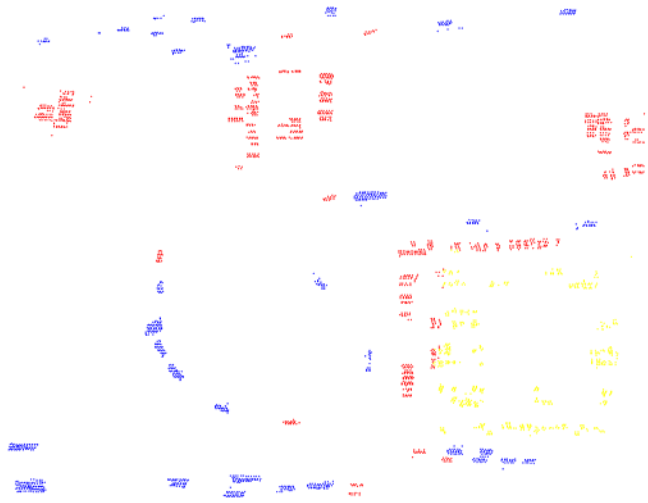
3.3 Results

We used the same vehicle datasets as derived in the section 2 to assess the proposed algorithm intended for classifying the motion status. Both of datasets are acquired over $300 \times 400\ m^2$ dense urban areas with averaged point density of about 1.4 pts/ m^2. The only one difference between them is that the first one used is co-registered from multiple strips rather than one-path. The classification results of vehicle motion status are presented in Fig.5. To access the performance of Lie group based classifier, minimum distance classifier was used to classify the same datasets based on the feature space spanned by vehicle parametrization.

The test dataset each consists of more than 50 vehicles successfully detected by vehicle extraction process. A set of 5 vehicle samples from each motion category is manually selected to train the classifier for vehicle motion status at first. It can be expected that poorly chosen training samples due to the strong shape variability in the category of moving vehicle could have a negative effect on classification performance. Therefore, the selection of training data for moving vehicle category should be carried out in such way that the fundamental shape information are expressed and generalized. Receiver Operating Characteristic (ROC) curves are generated by comparing classification results with reference data manually acquired by human interpretation and shown in Fig.6 for respective test datasets.

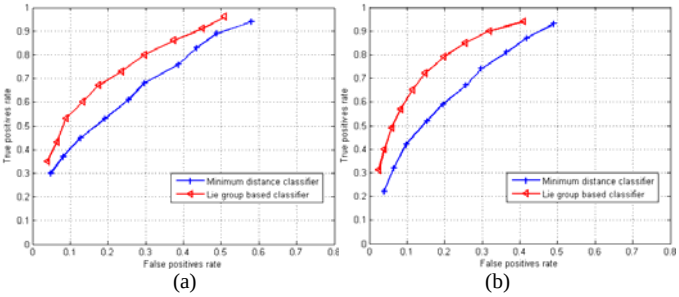


(a)



(b)

Figure 5. Vehicles motion classification results for dataset I and II (top-view of vehicle point sets). Blue: moving; Red: stationary; Yellow: uncertain.



(a) (b)

Figure 6. ROC curves for vehicle motion classification. (a) Dataset I; (b) Dataset II.

3.4 Discussion

Since we do not have real "ground truth" for vehicle motion which could be simultaneously captured along the scanning campaigns by an imaging sensor as described in Toth and Grejner-Brzezinska, (2006), the results are firstly assessed with respect to human examination abilities. Based on the context relations the vehicle movement could be roughly distinguished between moving vehicles and stationary ones. Note that the along-track motion cannot be resolved on principle if the true length is unknown, our evaluation are inherently biased by ambiguities introduced by the incorrect vehicle length.

It can be found out from the results displayed above that most of detected moving vehicles appear in the heavily travelled roads such as flyovers and main streets of city and the vehicles classified as motionless are mostly found in the parking lots or along road margins. The yellow class indicates the vehicles of uncertain status which are all nearly placed very close to each other in a parking lot and are excluded from the motion classification step due to the shape irregularity. False alarms from motion classification by our approach usually appear for slowly moving vehicles which travelled not perpendicular to the flight direction or those moving ones that are shaped by anomaly sample points in ALS data due to vegetation occlusion or unstable reflection properties. As indicated in ROC curves, the Lie group based classifier outperforms the minimum distance classifier in both cases, as its ability to generalize various shapes from training data, even for worst-cases, is demonstrated. It can also be observed that the second test

dataset generally has better performance than the first one in terms of vehicle motion classification, which has shown that one-path LiDAR data could be more appropriate for our task than co-registered data of multiple strips, despite that the point density of combined dataset would be higher. Moreover, the superior performance may trace back to the applied extraction strategy of direct 3D segmentation on LiDAR point clouds other than 2D analysis approach.

Once the motion status of extracted vehicles is determined, the velocity of moving vehicles can be inferred under the precondition that the true vehicle size is known. According to results presented here, it is easy to empirically give such performance summery that the vehicle motion indication as well as estimation from ALS data would fairly depend on certain factors, such as point density, distribution spacing between every two vehicles, relative motion direction to the flight direction, absolute velocity of vehicle, and vehicle size. The accurate impacts of single factors on motion analysis results have to be further obtained by quantitative analysis with great amount of test data

Traffic analysis could quite benefit from some distinctive operational conditions of LiDAR sensor, in comparison to optical camera. It is an active sensor less weather dependent; for example, it can cope with haze, fog and volume-scattering objects to some extent, working night too. Furthermore, scene complexity poses an additional difficulty for the optical imagery: dense urban areas, long and strong shadows, occlusions, etc., can severely impair the vehicle extraction performance.

4. CONCLUSION

Overall, a progressive scheme consisting of the vehicle extraction step followed by motion status classification is presented in this work attempting to automatically characterize the traffic scenario in urban areas. Based on single vehicle instances extracted by an approach combining context exploitation with 3D segmentation, the binary motion status of them is determined by shape analysis and classification. As indicted by the results derived from real ALS data commonly used for city mapping and modeling, traffic analysis by airborne LiDAR offers great potential to support the short/mid-term acquisition of statistical traffic data for a given road network in urban areas in despite of higher false alarm rates. Nevertheless, numerous potential improvements of the schemes have to be developed in future, in order to deal with main obstacles to LiDAR traffic characterization, especially regarding velocity estimation, such as low point density, unknown vehicle size and unstable laser reflection properties of vehicle surface.

REFERENCES

Fletcher, P.T., Conglin, L. and Joshi, S., 2003. Statistics of shape via principal geodesic analysis on Lie groups, Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on, pp. I-95-I-101 vol.1.

Hinz, S., Bamler, R. and Stilla, U. (Editors), 2006. Theme issue "Airborne and spaceborne traffic monitoring". ISPRS Journal of Photogrammetry and Remote Sensing, 61(3/4), 135-278 pp.

Hinz, S., Lenhart, D. and Leitloff, J., 2008. Traffic extraction and characterisation from optical remote sensing data. The Photogrammetric Record, 23(124): 424-440.

Jarvis, R.A., 1977. Computing the shape hull of points in the plane, IEEE Computing Society Conference on Pattern Recognition and Image Processing, New York, pp. 231–241.

Jin, X. and Davis, C.H., 2007. Vehicle detection from high-resolution satellite imagery using morphological shared-weight neural networks. Image and Vision Computing, 25(9): 1422-1431.

Meyer, F., Hinz, S., Laika, A., Weihing, D. and Bamler, R., 2006. Performance analysis of the TerraSAR-X Traffic monitoring concept. ISPRS Journal of Photogrammetry and Remote Sensing, 61(3-4): 225-242.

Reinartz, P., Lachaise, M., Schmeer, E., Krauss, T. and Runge, H., 2006. Traffic monitoring with serial images from airborne cameras. ISPRS Journal of Photogrammetry and Remote Sensing, 61(3-4): 149-158.

Rosenbaum, D., Kurz, F., Thomas, U., Suri, S. and Reinartz, P., 2008. Towards automatic near real-time traffic monitoring with an airborne wide angle camera system. European Transport Research Review.

Rossmann, W., 2002. 'Lie Groups: An introduction through linear groups. Oxford University Press.

Runge, H. et al., 2007. Space borne SAR traffic monitoring, Proceedings, International Radar Symposium, Cologne, pp. 5.

Sharma, G., Merry, C.J., Goel, P. and McCord, M., 2006. Vehicle detection in 1-m resolution satellite and airborne imagery. International Journal of Remote Sensing, 27(4): 779 - 797.

Stilla, U., Rottensteiner, F. and Hinz, S. (Editors), 2005. Object extraction for 3D city models, road databases, and traffic monitoring— concepts, algorithms, and evaluation (CMRT05) International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences, 36(3/W24), 196 pp.

Toth, C.K. and Grejner-Brzezinska, D., 2006. Extracting dynamic spatial data from airborne imaging sensors to support traffic flow estimation. ISPRS Journal of Photogrammetry and Remote Sensing, 61(3-4): 137-148.

Yao, W., Hinz, S. and Stilla, U., 2008a. Automatic vehicle extraction from airborne LiDAR data of urban areas using morphological reconstruction, Proceedings of 5th IAPR Workshop on Pattern Recognition in Remote Sensing (PRRS08), Tampa, USA, pp. 1-4.

Yao, W., Hinz, S. and Stilla, U., 2008b. Traffic monitoring from airborne LIDAR – Feasibility, simulation and analysis, XXI Congress, Proceedings. International Archives of Photogrammetry, Remote Sensing and Spatial Geoinformation Sciences, Beijing, China, pp. Vol 37(B3B):593-598.

Yao, W., Hinz, S. and Stilla, U., 2009. Object extraction based on 3d-segmentation of LiDAR data by combining mean shift and normalized cuts: two examples from urban areas, 2009 Urban Remote Sensing Joint event: URBAN 2009 - URS 2009.

Yarlagadda, P., Ozcanli, O. and Mundy, J., 2008. Lie group distance based generic 3-d vehicle classification, Pattern Recognition, 2008. ICPR 2008. 19th International Conference on, pp. 1-4.

Zhao, T. and Nevatia, R., 2003. Car detection in low resolution aerial images. Image and Vision Computing, 21(8): 693-703.

TRAJECTORY-BASED SCENE DESCRIPTION AND CLASSIFICATION BY ANALYTICAL FUNCTIONS

D. Pfeiffer [a], R. Reulke [b*]

[a] Akeleiweg 46D, 12487 Berlin, Germany - david@dltmail.de
[b] Humboldt-University of Berlin, Computer Science Department, Computer Vision, Rudower Chaussee, Berlin, Germany - reulke@informatik.hu-berlin.de

Commission III, WG III/4 and III/5

KEY WORDS: Object recognition, Scene Analysis, statistical and deterministic trajectory models

ABSTRACT:

Video image detection systems (VIDS) provide an opportunity to analyse complex traffic scenes that are captured by stationary video cameras. Our work concentrates on the derivation of traffic relevant parameters from vehicle trajectories. This paper examines different procedures for the description of vehicle trajectories using analytical functions. Derived conical sections (circles, ellipses and hyperboles) as well as straight lines are particularly suitable for this task. Thus, it is possible to describe a suitable trajectory by a maximum of five parameters. A classification algorithm uses these parameters and takes decisions on the turning behaviour of vehicles.
A model based approach is following. The a-priori knowledge about the scene (here prejudged and verified vehicle trajectories) is the only required input into this system. One confines himself here to straight lines, circles, ellipses and hyperboles. Other common functions (such as clothoids) are discussed and the choice of the function is being justified.

1. INTRODUCTION

1.1 Motivation

Traffic management is based on an exact knowledge of the traffic situation. Therefore, traffic monitoring at roads and intersections is an essential prerequisite. Inductive loops and microwave radar systems are the most common detection and surveillance systems to measure traffic flow on public roads.

VIDS that operate with real time image processing techniques became more attractive during the last 15 years (Michalopoulos 1991), (Wigan 1992), (Setchell et al. 2001), (Kastrinaki et al. 2003). Traditional traffic parameters like presence, vehicle length, speed as well as time gap between two vehicles and vehicle classification (Wei et al. 1996) can be determined. In contrast to other sensors, the use of local cameras makes a two-dimensional observation possible and thus can determine new traffic parameters like congestion length, source-destination matrices, blockage or accidents and therefore support the estimation of travel times. Multi-camera systems extend some limitations of single camera systems (e.g. occlusions, reliability) and enlarge the observation area (Reulke et al. 2008a).

We proposed a framework that autonomously detects atypical objects, behavior or situations even in crowded and complex situations (Reulke et al. 2008b). Extracted object data and object trajectories from multiple sensors have to be fused. An abstract situational description of the observed scene is obtained from the derived trajectories. The first step in describing a traffic scene is to ascertain the normal situation by statistical means. In addition, semantic interpretation is also derived from statistical information (such as direction and speed). Deviations of the inferred statistics are interpreted as atypical events, and therefore can be used to detect and prevent dangerous situations. These options allow the detection of sudden changes as well as atypical or threatening events in the scene. Atypical or threatening events are generally defined as deviations from the normal scene behavior or have to be defined by a rule based scheme. Red light runners and incident detection systems are an example for a self-evident road traffic application.

The trajectories of street vehicles are smooth and homogeneous over a large scale. Therefore, a mathematical description by elementary functions is appropriate for these trajectories. Thus, dramatic reductions of the bandwidths are achieved for a full scene transmission. The basic step to determine the driver intentions is to fit the trajectories to the analytical functions.

This paper is organized as follows: After an overview of situation analysis and atypical event detection the approach is introduced. Then, an example installation is described and its results are presented. The mathematical fundamentals of the adaptation of formerly derived trajectories of turning vehicles by hyperbolas, ellipsoids, spheres and straight lines are sketched. The derived information is very comprehensive but compact and permits downcast to other representations like source destination matrices. The paper closes with a summary and an outlook.

1.2 Situation Analysis and Atypical Event Detection

Scene description and automatic atypical event detection are issues of increasing importance and an interesting topic in many scientific, technical or military fields where complex situations (i.e. scenes containing many objects and interac-

* Corresponding author.

tions) are observed and evaluated. A common aim is to describe the observed data and to detect atypical or threatening events.

Other areas of situation analysis besides driver assistance (Reichardt 1995) may include traffic situation representation, surveillance applications (Beynon et al. 2003), sport video analysis or even customer tracking for marketing analysis (Leykin et al. 2005).

(Kumar et al. 2005) developed a rule-based framework for behavior and activity detection in traffic videos obtained from stationary video cameras. For behavior recognition, interactions between two or more mobile targets as well as between targets and stationary objects in the environment have been considered. The approach is based on sets of predefined behavior scenarios, which need to be analyzed in different contexts.

(Yung et al. 2001) demonstrate a novel method for automatic red light runner detection. It extracts the state of the traffic lights and vehicle motions from video recordings.

1.3 Image and Trajectory Processing

The cameras deployed cover overlaid or adjacent observation areas. With it, the same road user can be observed using different cameras from different view positions and angles. The traffic objects in the image data can be detected using image processing methods.

The image coordinates of these objects are converted to a common world coordinate system in order to enable the tracking and fusion of the detected objects of the respective observation area. High precision in coordinate transformation of the image into the object space is required to avoid misidentification of the same objects that were derived from different camera positions. Therefore, an exact calibration (interior orientation) as well as knowledge of the position and view direction (exterior orientation) of the camera is necessary.

Since the camera positions are given in absolute geographical coordinates, the detected objects are also provided in world coordinates.

The approach is subdivided into the following steps. Firstly, all moving objects have to be extracted from each frame of the video sequences. Secondly, these traffic objects have to be projected onto a geo-referenced world plane. Afterwards, these objects are tracked and associated to trajectories. One can now utilize the derived information to assess comprehensive traffic parameters and to characterize trajectories of individual traffic participants.

1.4 Scenario

The scenario has been tested at the intersection Rudower Chaussee / Wegedornstrasse, Berlin (Germany) by camera observation using three cameras mounted at a corner building at approximately 18 meters height. The observed area has an extent of about 100x100 m and contains a T-section. Figure 1 shows example trajectories derived from images taken from three different positions. The background image is an orthophoto, derived from airborne images.

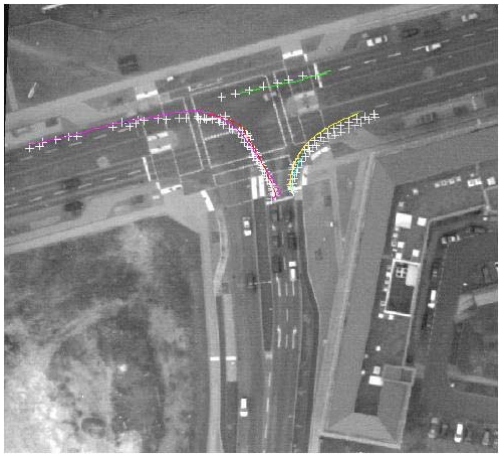


Figure 1. Orthophoto with example trajectories

The aim is the description of the trajectories by functions with a limited number of parameters. Source destination matrices could be determined at these crossroads through such parameters without any further effort. A classification approach shall be used here.

2. PROCESSING APPROACH

2.1 Video Acquisition and Object Detection

In order to receive reliable and reproducible results, only compact digital industrial cameras with standard interfaces and protocols (e.g. IEEE1394, Ethernet) are deployed.

Different image processing libraries or programs (e.g. OpenCV or HALCON) are available to extract moving objects from an image sequence. We used a special algorithm for background estimation, which adapts to the variable background and extracts the desired objects. The dedicated image coordinates as well as additional parameters like size and area were computed for each extracted traffic object.

2.2 Sensor Orientation

The existing tracking concept is based on extracted objects, which are geo-referenced to a world coordinate system. This concept allows the integration or fusion of additional data sources. The transformation between image and world coordinates is based on collinearity equations. The Z-component in world coordinates is deduced by appointing a dedicated ground plane. An alternative is the use of a height profile. Additionally needed input parameters are the interior and exterior orientation of the camera. For the interior orientation (principal point, focal length and additional camera distortion) of the cameras the 10 parameter Brown distortion model (Brown 1971) was used. The parameters are being determined by a bundle block adjustment.

Calculating the exterior orientation of a camera (location of the projection centre and view direction) in a well known world coordinate system is based on previously GPS measured ground control points (GCPs). The accuracy of the points is better than 5 cm in position and hight. The orientation is deduced through these coordinates using DLT and the spatial resection algorithm (Luhmann 2006).

2.3 Tracking and Trajectories

The aim of tracking is to map observations of measured objects to existing trajectories and to update the state vector describing those objects, e.g. position or shape. The tracking is carried out using a Kalman-filter approach.
The basic idea is to transfer supplementary information concerning the state into the filter approach in addition to the measurement. This forecast of the measuring results (prediction) is derived from earlier results of the filter. Consequently, this approach is recursive.

The initialization of the state-vector is conducted from two consecutive images. The association of a measurement to an evaluated track is a statistical based decision-making process. Errors are related to clutter, object aggregation and splitting. The decision criteria minimize the rejection probability.

The coordinate projection mentioned in the last paragraph and the tracking process provides the possibility to fuse data acquired from different sensors. The algorithm is independent of the sensor as long as the data is referenced in a joint coordinate system and they share the same time frame.

The resulting trajectories are then used for different applications e.g. for the derivation of traffic parameters (TP).

2.4 Trajectory analysis

A deterministic description method for trajectories shall be introduced below. The functional descriptions for these trajectories should be as simple as possible and permit a straightforward interpretation. Linear movements will be described by simple straight lines.

Numerous suggestions of possible functions for curve tracks by functional dependencies have been made in the literature. Clothoid (Liscano et al. 1989) or G2-Splines (Forbes 1989) are curves whose bend depends of the arc length. Alternatively, closed functions like B-Splines, Cartesian polynomials fifth degree or Polarsplines (Nelson 1989) can be used as well. A common approach to approximate vehicle-based trajectories is to employ clothoids. Those functions derived from the fresnel integral are highly non linear. They are fundamental in road and railroad construction. Due to urban constraints the tracks of intersections and curves cannot follow the curve of a clothoid whose shape is regarded as a trajectory that is especially comfortable to drive. Because there are only partial approximations of clothoids, they do not fit into the set of elementary functions that shall be regarded in this work. Moreover, the given trajectory has to be subdivided into parts in order to apply a clothoidal approximation. (Anderson et al. 1979) have proposed a description of tracks by hyperbolas. The great advantage is that the derived parameters clarify directly geometric connections and permit a categorization and derivation of important features of the trajectories. A hyperbola is able to replicate straight lines as well as turning trajectories.

The hyperbola fit serves as an example and is described next. The approach is based on least-square fitting of geometric elements. The equation for a hyperbola with semi-major axis parallel to the x-axis and semi-minor axis b parallel to the y-axis is given by

$$\frac{x^2}{a^2}-\frac{y^2}{b^2}=1 \tag{1}$$

The parametric equation is given by

$$x = a\cdot\sec(t) \quad y = b\cdot\tan(t) \tag{2}$$

Commonly the hyperbola is rotated and shifted:

$$\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} \cos\varphi & \sin\varphi \\ -\sin\varphi & \cos\varphi \end{pmatrix} \cdot \begin{pmatrix} x'-x_m \\ y'-y_m \end{pmatrix} \tag{3}$$

Wherein m_x,, m_y are the centre coordinates, the angle φ is the bearing of the semi-major axis. The implicit form of the hyperbola can be written as a general polynomial of second degree:

$$a_1'\cdot x'^2 + a_2'\cdot x'y' + a_3'\cdot y'^2 + a_4'\cdot x' + a_5'\cdot y' = 1 \tag{4}$$

With

$$\begin{aligned} a_1' &= \frac{a_1}{a_4} \quad a_2' = \frac{a_2}{a_4} \quad a_3' = \frac{a_3}{a_4} \\ a_4' &= -\frac{2\cdot x_m a_1 + y_m a_2}{a_4} = -\left(2\cdot x_m a_1' + y_m a_2'\right) \\ a_5' &= -\frac{x_m a_2 + 2\cdot y_m a_3}{a_4} = -\left(x_m a_2' + 2\cdot y_m a_3'\right) \end{aligned} \tag{5}$$

and

$$\begin{aligned} a_1 &= \frac{\cos^2\varphi}{a^2} - \frac{\sin^2\varphi}{b^2} \\ a_2 &= 2\cdot\sin\varphi\cdot\cos\varphi\cdot\left(\frac{1}{a^2}+\frac{1}{b^2}\right) \\ a_3 &= \frac{\sin^2\varphi}{a^2} - \frac{\cos^2\varphi}{b^2} \\ a_4 &= 1-\left(x_m^2 a_1 + x_m y_m a_2 + y_m^2 a_3\right) \end{aligned} \tag{6}$$

The following equations describe the conversion of the implicit to the hyperbola parametric form:

- Bearing of the semi-major axis

$$\varphi = a\tan\frac{a_2'}{a_1'-a_3'}\Big/2 \tag{7}$$

- Center coordinates

$$\begin{aligned} x_m &= -\frac{2\cdot a_4' a_3' - a_5' a_2'}{4\cdot a_1' a_3' - a_2'^2} \\ y_m &= -\frac{2\cdot a_1' a_5' - a_4' a_2'}{4\cdot a_1' a_3' - a_2'^2} \end{aligned} \tag{8}$$

- Semi-major axis

$$a^2 = \frac{1-\left(a'_4 \cdot \frac{x_m}{2} + a'_5 \cdot \frac{y_m}{2}\right)}{a'_1 \cdot \cos^2\varphi + a'_2 \cdot \sin\varphi \cdot \cos\varphi + a'_3 \cdot \sin^2\varphi} \qquad (9)$$

$$b^2 = -\frac{1-\left(a'_4 \cdot \frac{x_m}{2} + a'_5 \cdot \frac{y_m}{2}\right)}{a_3 \cdot \cos^2\varphi - a_2 \cdot \sin\varphi \cdot \cos\varphi + a_1 \cdot \sin^2\varphi}$$

The parameter determination is based on the number of observations n, which are related to the functional model. The number of observations n has to be greater than the number of unknown parameters.

$$a'_1 \cdot x'^2_i + a'_2 \cdot x'_i y'_i + a'_3 \cdot y'^2_i + a'_4 \cdot x'_i + a'_5 \cdot y'_i = 1 \qquad (10)$$

or

$$\begin{bmatrix} x'^2_0 & x'_0 y'_0 & y'^2_0 & x'_0 & y'_0 \\ x'^2_1 & x'_1 y'_1 & y'^2_1 & x'_1 & y'_1 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x'^2_{n-1} & x'_{n-1} y'_{n-1} & y'^2_{n-1} & x'_{n-1} & y'_{n-1} \end{bmatrix} \cdot \begin{bmatrix} a'_1 \\ a'_2 \\ a'_3 \\ a'_4 \\ a'_5 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} \qquad (11)$$

This can also be written as follows:

$$\underline{l} = \underline{\underline{A}} \cdot \underline{x} \qquad (12)$$

The observation vector $\underline{l}$ is replaced by the measured observation and a small residuum ν. Therefore, the unknown vector x is replaced by the estimates with the result:

$$\underset{u}{\hat{x}} = \left(\underset{u,n}{\underline{\underline{A}}^T} \cdot \underset{n,n}{\underline{\underline{P}}} \cdot \underset{n,u}{\underline{\underline{A}}} \right)^{-1} \cdot \underset{u,n}{\underline{\underline{A}}^T} \cdot \underset{n,n}{\underline{\underline{P}}} \cdot \underset{n}{\underline{l}} \qquad (13)$$

This result is known as a least-square adjustment, based on the L_2 norm. This approach is not able to decide between hyperbola and ellipsoid. (Fitzgibbon et al. 1996) and (Fitzgibbon et al. 1999) describe an attempt for the inclusion of additional conditions by integration of a constraint matrix. Hence it is possible to reduce the resulting solution space so that the type of the object function (ellipse, hyperbole) can be steered. (Harlow et al. 2001) and (Harker et al. 2008) enlarge Fitzgibbon's approach by decomposition of the Scattermatrix in the square, linear and constant part. The parameter estimate becomes equivalent to the eigenvalue problem. This is a direct solution method. The approach determines an ellipse as well as two hyperboles. Figure 2 shows examples for the hyperbola and ellipse fit.

Figure 2. Example fits for two tracks and classification

3. CLASSIFICATION

3.1 Class definition

The traffic objects are identified within the image data and trajectories are derived from it. The trajectories are fitted to curves and their parameters are classified with the corresponding functions. For the classification the same data set is used for all function classes.

A part of the data set is used to train a classifier which intends a class assignment for the trajectory with the parameters. The other part serves for the verification.

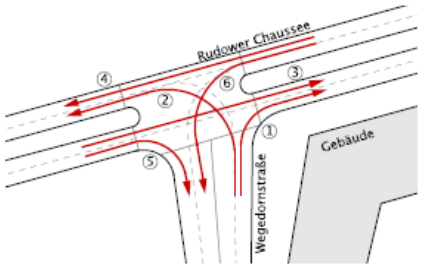


Figure 3. Visualisation of the different traffic lanes (classes) of the scene

Seven classes were defined based on the scene (figure 3) and the traffic lanes:

No	From	To	Class	Abbreviation
1	Wegedorn	Rudower	right-turn	WRR
2	Wegedorn	Rudower	left-turn	WRL
3	Rudower	Rudower	east-direction	RO
4	Rudower	Rudower	west-direction	RW
5	Rudower	Wegedorn	right-turn	RWR
6	Rudower	Wegedorn	left-turn	RWL
7	No class membership			No_Class

Table 1. Class definition for the observed scene

The used data set consists of 414 trajectories. Trajectories which are part of the classification process need to have a minimal length of 10 m and a minimal number of points of at least 6 points. The class No_Class consist of trajectories of pedestrians, bicyclists and erroneous tracks caused by errors in image processing and tracking. It is inadmissible that two driving directions are assigned to one trajectory. Relying on the shape only, opposite directions is merely to distinguish, since their functional parameters are similar. To achieve the distinction the approximate same path of the trajectory and the fitted function is determined. With this, the direction of the trajectory can be determined as an additional feature. Hence it direction can be distinguished between close lanes trajectories with opposite directions.

3.2 Classification method

A classifier determines the class affiliation with the characteristic of item-specific features. These features are represented as a vector in a multidimensional feature space. The features correspond to the parameters which have been determined by the approximation.

A rectangle classifier and a modified k nearest neighbours (KNN-) classifier are used. The result of the classification shall be unambiguously.

k-nearest neighbours algorithm (KNN) is a method for classifying objects based on closest training examples in the feature space.

The rectangle classification (also cuboid classification) is a distribution free, nonparametric and supervised classification method (see figure 4).

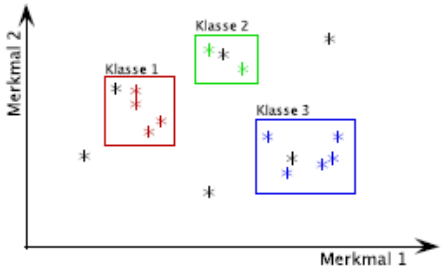


Figure 4. A simple rectangle classification in a 2D feature space

The KNN classification needs a training data set. It is a non-parametric method for the estimate of probability densities. The operation of the classifier is steered by k (number of regarded neighbours, a free selectable parameter) and δ (used metric). Figure 5 shows the approach.

Figure 5. Visualization of the KNN classification. The k=7 nearest neighbour are used. The object g is assigned to the class B

The metric δ defines the reliable determination of the distances to adjacent elements. The result of the classification depends substantially on the density of the learning set and the choice of the metric. Here the Mahalanobis distance was used.

4. RESULTS

	Total	NC	WRR	WRL	RO	RW	RWR	RWL
Ref	414	62	117	59	72	26	33	54
Circ	414	62	119	59	68	21	34	51
Elli	410	51	117	58	72	28	34	50
Hyp	410	51	117	58	72	28	34	50
Str	413	50	125	56	70	28	35	49

Table 2. Summary of complete occurrence and the class occurrence of different trajectory types. Ref – reference, Circ – circle, Elli – ellipse, Hyp – hyperbola, Str - straight lines

A data set of 414 different trajectories (Total) has been processed using different functions within the test data set. A total of 62 trajectories could not be classified (NC). A summary is given in table 2.

The results shall be represented in greater detail by the hyperboles in the following.

4.1 Hyperbola

Figure 6 shows examples of the approximation of hyperboles.

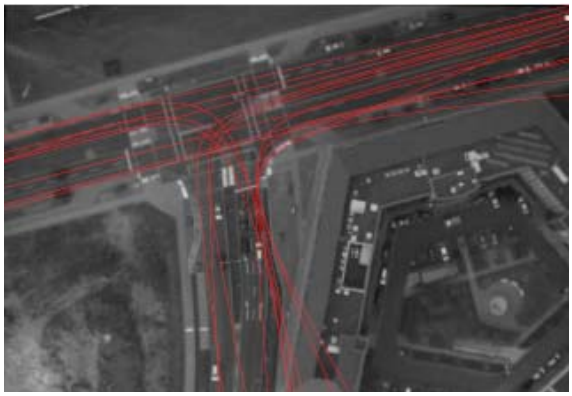


Figure 6. Approximation of hyperboles

In addition to the parameters of the conical sections the direction of motion was uses for the classification. Figure 7 shows the plot of the rotational angle φ (X) and the delta in degrees (ϕ) where the trajectory adapts to the hyperbola:

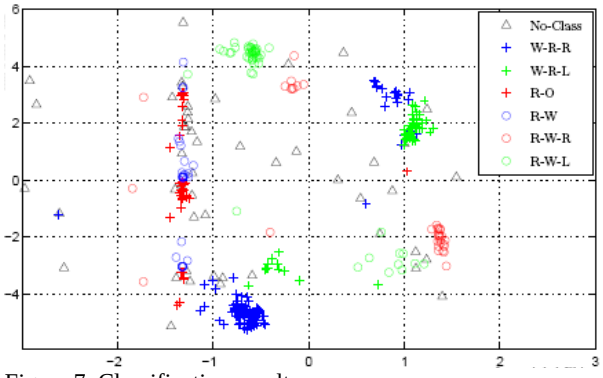


Figure 7. Classification results

Class	Cuboid Classifier	KNN-Classifier
Total	92.9%	97.8%
No_Class	84.3%	98.0%
WRR	95.7%	99.1%
WRL	93.1%	99.0%
RO	97.2%	99.5%
RW	96.4%	96.4%
RWR	85.3%	91.2%
WRL	92.00%	94.0%

Table 3. Comparison of results for hyperbolas fits achieved by Coboid and KNN-Classifier

Figure 7 shows a clear separation of the feature space. A high classification rate is achieved by both elementary classifiers (see table 3).

5. CONCLUSION AND OUTLOOK

Table 3 affirms a high reliability on these elementary functions, with respect to the used basic classification methods. Mistakes within the classification mostly reside due to scene behaviour that occurs fairly rare (e.g. car turning at the intersection) or is not modelled by the underlying functions (e.g. pedestrians or cyclists crossing in very custom patterns). The shown approaches have been tested and verified in a real-time environment with a multi-camera system.

The system shall to automatically observe the traffic on crossroads in future. For example source-destination dependences can be determined with that.

6. REFERENCES

Anderson, B. and J. Moor (1979). Optimal filtering. Enlewood Cliffs, New Jersey, Prentice-Hall, Inc.

Beynon, M. D., D. J. V. Hook, et al. (2003). Detecting Abandoned Packages in a Multi-Camera Video Surveillance System. IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS'03).

Brown, D. C. (1971). "Close range camera calibration." Photogrammetric Engineering **37**(8): 12.

Fitzgibbon, A., M. Pilu, et al. (1999). "Direct least square fitting of ellipses." Pattern Analysis and Machine Intelligence, IEEE Transactions on **21**(5): 476-480.

Fitzgibbon, A. W., M. Pilu, et al. (1996). Direct least squares fitting of ellipses. Pattern Recognition, 1996., Proceedings of the 13th International Conference on.

Forbes, A. B. (1989). Least-squares best fit geometric elements, National Physical Laboratory of Great Britain.

Harker, M., P. O'Leary, et al. (2008). "Direct type-specific conic fitting and eigenvalue bias correction." Image Vision Comput. **26**(3): 372-381.

Harlow, C. and Y. Wang (2001). "Automated Accident Detection System." Transportation Research Record **1746**(-1): 90-93.

Kastrinaki, V., M. Zervakis, et al. (2003). "A survey of video processing techniques for traffic applications." Image and Vision Computing **21**(4): 359-381.

Kumar, P., S. Ranganath, et al. (2005). "Framework for real-time behavior interpretation from traffic video." Intelligent Transportation Systems, IEEE Transactions on **6**(1): 43-53.

Leykin , A. and M. Tuceryan (2005). A Vision System for Automated Customer Tracking for Marketing Analysis: Low Level Feature Extraction. International Workshop on Human Activity Recognition and Modelling. Oxford, UK.

Liscano, R. and D.Green (1989). Design and implementation of a trajectory generator for an indoor mobile robot. Proceedings of the IEEE/RJS International Conference on Intelligent Robots and Systems. Tsukuba, Japan**:** 380–385.

Luhmann, T. (2006). "Close Range Photogrammetry: Principles, Methods and Applications."

Michalopoulos, P. G. (1991). "Vehicle detection video through image processing: the Autoscope system." Vehicular Technology, IEEE Transactions on **40**(1): 21-29.

Nelson, W. L. (1989). "Continuous steering-function control of robot carts." IEEE Transactions on Industrial Electronics **36**(3): 330–337.

Reichardt, D. (1995). A Real-time Approach to Traffic Situation Representation from Image Processing Data. Intelligent Vehicles Symposium. Detroit, MI, USA.

Reulke, R., S. Bauer, et al. (2008a). Multi-Camera Detection and Multi-Target Tracking. VISAPP 2008. Funchal, Madeira (Portugal).

Reulke, R., S. Bauer, et al. (2008b). Situation Analysis and Atypical Event Detection with Multiple Cameras and Multi-Object Tracking. Robot Vision. Auckland (New Zeeland), Springer-Verlag Berlin Heidelberg.

Setchell, C. and E. L. Dagless (2001). "Vision-based road-traffic monitoring sensor." Vision, Image and Signal Processing, IEE Proceedings - **148**(1): 78-84.

Wei, C.-H., C.-C. Chang, et al. (1996). "Vehicle Classification Using Advanced Technologies." Transportation Research Record **1551**(-1): 45-50.

Wigan, M. R. (1992). "Image-Processing Techniques Applied to Road Problems." Journal of Transportation Engineering **118**(1): 21.

Yung, N. H. C. and A. H. S. Lai (2001). "An effective video analysis method for detecting red light runners." Vehicular Technology, IEEE Transactions on **50**(4): 1074-1084.

3D BUILDING RECONSTRUCTION FROM LIDAR BASED ON A CELL DECOMPOSITION APPROACH

Martin Kada[a], Laurence McKinley[b]

[a] Institute for Photogrammetry, University of Stuttgart, Geschwister-Scholl-Str. 24D, 70174 Stuttgart, Germany
martin.kada@ifp.uni-stuttgart.de
[b] Virtual City Systems, Zellescher Weg 3, 01069 Dresden, Germany
lmckinley@virtualcitysystems.de

Commission III, WG III/4

KEY WORDS: LIDAR, Reconstruction, Building, Automation, Algorithms

ABSTRACT:

The reconstruction of 3D city models has matured in recent years from a research topic and niche market to commercial products and services. When constructing models on a large scale, it is inevitable to have reconstruction tools available that offer a high level of automation and reliably produce valid models within the required accuracy. In this paper, we present a 3D building reconstruction approach, which produces LOD2 models from existing ground plans and airborne LIDAR data. As well-formed roof structures are of high priority to us, we developed an approach that constructs models by assembling building blocks from a library of parameterized standard shapes. The basis of our work is a 2D partitioning algorithm that splits a building's footprint into nonintersecting, mostly quadrangular sections. A particular challenge thereby is to generate a partitioning of the footprint that approximates the general shape of the outline with as few pieces as possible. Once at hand, each piece is given a roof shape that best fits the LIDAR points in its area and integrates well with the neighbouring pieces. An implementation of the approach is used now for quite some time in a production environment and many commercial projects have been successfully completed. The second part of this paper reflects the experiences that we have made with this approach working on the 3D reconstruction of the entire cities of East Berlin and Cologne.

1. INTRODUCTION

3D building reconstruction has been a topic for quite some time now. Many research papers have been published; commercial services and software are available. (Brenner, 2005), e.g., gives a good overview of reconstruction methods and points out that "research is still far from the goal of the initially envisioned fully automatic reconstruction systems". This situation has not yet changed much, although a lot of research is still devoted to this topic, as can be seen in the multitude of recent publications (e.g. (Arefi et al., 2008), (Möser et al., 2009), (Sohn et al., 2008)).

The subject of this paper is on the generation of realistic 3D city models in LOD2 as it is defined in the official OGC standard CityGML (see e.g. (Kolbe, 2009)). At this LOD, buildings have distinctive roof structures and flat facades that are textured from terrestrial or oblique aerial images.

As the data basis, we rely on existing ground plans and airborne LIDAR data. A frequent requirement, especially from customers within the mainland Europe, is that the provided building outlines are to be preserved with only little tolerance and that ridge and eaves heights must be very accurate. This is especially important so that the facades and roofs can be properly mapped from oblique aerial images.

The presented reconstruction approach is motivated from our research on the simplification of 3D building models for map-like representations (Kada, 2007). An integral part of this work lies on a new method to decompose a 2D building footprint into a small set of nonintersecting primitives. Although the resulting partitioning only approximates the original outline, it is still accurate enough for reconstruction purposes. The benefit is, however, that the algorithm separates the sections nicely, especially for residential houses with gabled or hipped roofs. This eases the task of determining and assembling a valid roof structure from parameterized, standard shapes.

In the second part of the paper, we give insight into two large-area projects that we have completed using the described 3D reconstruction system: East Berlin and Cologne. Figure 1 shows the reconstructed 3D city model of Berlin with textures mapped from oblique imagery.

Figure 1. Real-time visualization of the 3D city model of Berlin.

2. RECONSTRUCTION ALGORITHM

In our approach, we assume that the majority of residential houses have either one main section or multiple connected sections, with additional smaller extensions, and that a partition thereof can be properly derived from the outline polygon. Once such a partition is found, a general geometrical description of the roof can be constructed by assigning a parameterized standard shape to each section. However, the difficulty to generate correct facade and roof shapes from a partition increases with the number, shape and arrangement of its elements. We therefore generate a set of non-overlapping, mostly quadrilateral shaped polygons that together approximate the original footprint (cp. Figure 2). Other ground shapes may also occur, but those primitives are then restricted to only bear certain roof shapes.

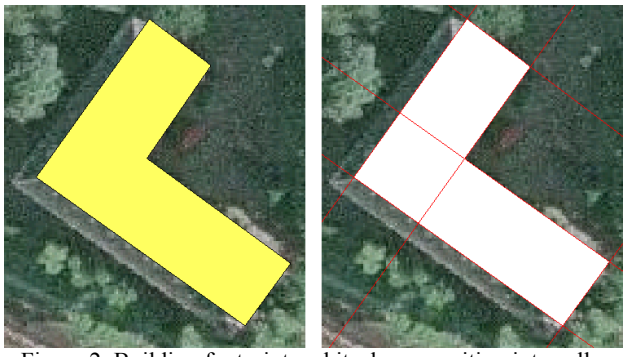


Figure 2. Building footprint and its decomposition into cells.

The roof is then reconstructed by determining a shape for each cell from the LIDAR points with regard to the neighbour cells (cp. Figure 3). After identifying the points inside a cell, the normal vectors from the local regression planes of the points are tested against all possible shapes. Here, only the orientation is used to speed up comparing the many shapes we support. The one that best fits is then chosen and its parameters estimated from the 3D point coordinates. Cells whose neighbour configurations suggest corner-, t- and cross-junctions are examined again and replaced if a junction shape can be fitted according to the neighbour shapes and parameters.

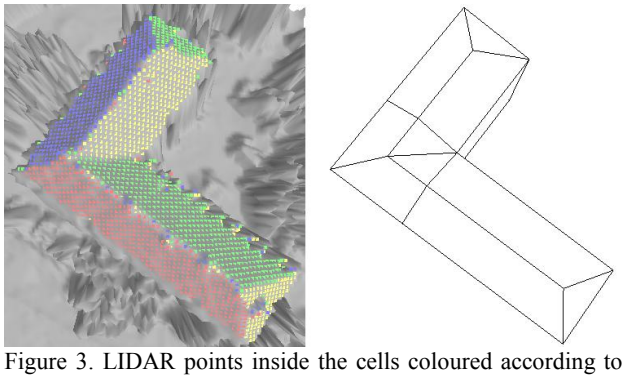


Figure 3. LIDAR points inside the cells coloured according to their local regression plane and the best fitting roof shapes.

After the geometric reconstruction, the building models are textured from oblique aerial images. Any lack of geometric detail that is due to our rather restricting model oriented approach is then hardly noticeable in the result.

2.1 Cell Decomposition

As referred to in (Foley et. al, 1996), a spatial partitioning representation in solid modelling, where solids are decomposed into nonintersecting, typically parameterized primitives, is called cell decomposition.

Serving as the basis for the building reconstruction process, we first of all generate such a partition for each building footprint. As mentioned above, this is done solely from information found in the building's outline. The big challenge herein is to avoid decomposing the area in too many small cells, for which it becomes increasingly difficult to reconstruct a well-shaped roof, especially if the building outline is very detailed and consists of many short line sections (see Figure 4). So instead of using all the available lines from the outline polygon and infinitely extend them to split the footprint, an adequate subset must be found that results in a set of primitives that together reflects well the characteristic shape of the building. However, the resulting outline will not be identical to the original one, but rather be a generalization thereof. So to best resemble the outline, the set of decomposition lines should approximate well the original points and line segments.

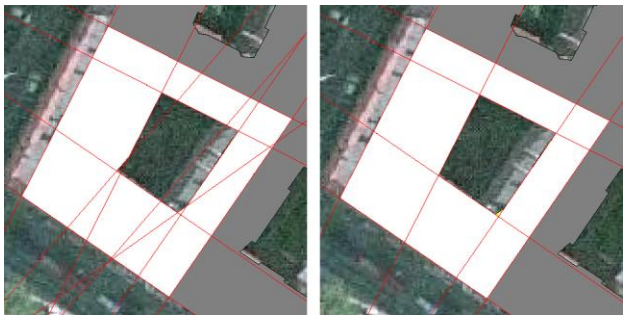


Figure 4. Cell decomposition of a building footprint using all line segments of the outline and only an averaged subset.

Our algorithm for generating cell decompositions from given outlines has been thoroughly described in the context of 3D building generalization (see e.g. (Kada, 2007)). But instead of generating 3D decomposition planes from the facade polygons of a 3D building model, the 2D decomposition lines are now generated from the 2D outline.

In a nutshell, the line segments are grouped into subsets of "parallel" lines that are pair wise a maximum distance away from each other. This is the generalization distance, which means in this context, that the cells resulting from the footprint partitioning will not have sides that are shorter than this length. Line segments are considered parallel if the angle between their directions is below an angle threshold. This allows for a better generalization of connected line segments and therefore helps to keep the number of generated cells low. For each subset of line segments, the associated decomposition line is computed by averaging the line equations of its elements. Short line segments of arbitrary direction, but whose endpoints are both closer to the decomposition line than the parallel line segments, are associated with this subset, but will not contribute to the averaging of this or any other decomposition line.

For example, the green line segments on the left side of Figure 5 are considered parallel under the chosen angle threshold of 15 degrees. The added perpendicular distance of any two endpoints

to the red decomposition line, which is the average of the green line segments, is below the generalization distance. While the connecting orange line segment is not parallel to any green line segments, its endpoints also falls under the distance threshold and therefore does not contribute to any decomposition line.

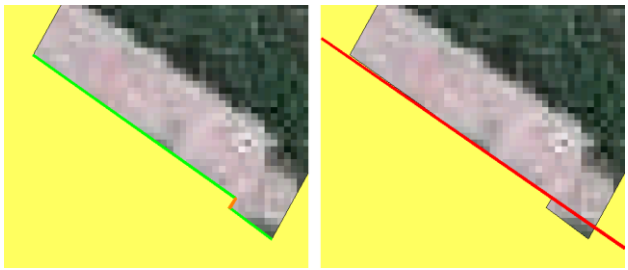


Figure 5. Parallel line segments (green) form decomposition lines (red), rendering short segments in between (orange) unnecessary.

Under the general assumption that ridge and eaves lines should strictly run horizontally, many roof shapes require the ground shape of cells to be trapezoids or rhomboids. Otherwise not all roof faces will be planar and must be split into triangles to form valid solids. Figure 6 shows an extreme example of a cell with a Berliner roof shape where none of the four sides of the ground shape are parallel. The middle face of the roof must be split into two triangles, which is generally not acceptable and should be avoided if possible. Due to the averaging process, the set of resulting decomposition lines are not guaranteed to be parallel. We therefore adjust the decomposition lines slightly so that parallelism and rectangularity are enforced for pairs of decomposition lines with small directional deviations. The same Berliner roof shape of Figure 6 with a trapezoid ground shape results in a valid solid after adjustment.

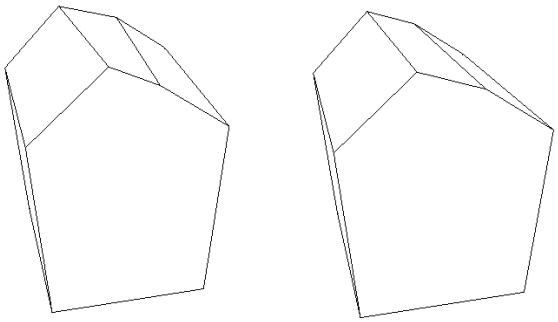


Figure 6. Extreme example of a Berliner roof primitive with a non-parallel before and a trapezoid roof shape after adjustment.

Once the decomposition lines have been generated, a rectangle approximately two times the minimal bounding rectangle is taken and split by these lines, forming nonintersecting cells in the process. Then the cells are compared with the original footprint, and the ones with a low overlap value are discarded. Large cells assure that this classification fails only in few cases.

Figure 7 shows an example cell decomposition of a given footprint. Cells with a low overlap with the original footprint were discarded in the process. The four "horizontal" lines are pair wise parallel, whereas the five "vertical" lines are all parallel, resulting in mostly rhomboid-shaped cells. Although the dotted cells are shaped as trapezoids, most roof shapes fitting between two opposite neighbour cells are valid under these conditions.

Figure 7. Cell decomposition of a given footprint into rhomboids and trapezoids, the latter marked with dots.

2.2 Roof Shape Determination

Now that a cell decomposition of the footprint is available, the parameterized roof shapes of all cells need to be found. We do this by examining the normal vectors of all points inside the same cell. As point normal vectors are usually not given in surface models, they first have to be generated. If the surface model is structured as a grid, we compute the normal vector of each point from the eight triangles fanned around it and average their normal vectors. However, if the raw data is available in form of an unstructured point cloud, we estimate a point's local plane of regression from its five nearest neighbours and take the resulting surface normal vector.

For the construction of the building's roof, we classify the roof shapes that we use in our approach into three types: basic, connecting and manual shapes. Whereas the shapes of the first two classes can be determined in an automatic process, the last class of roof shapes is only available for manual editing. Among the basic roof shapes are flat, shed, gabled, hipped and Berliner roof (see Figure 8).

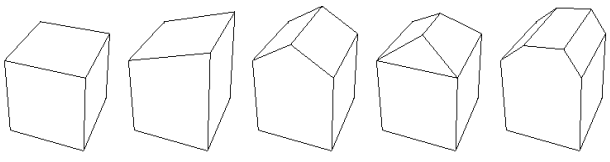


Figure 8. Flat, shed, gabled, hipped and Berliner roof shape.

As not all houses have only one section, there is a need to connect the roofs of the sections with specific junction shapes. Figure 9 shows a small selection of connecting roof shapes.

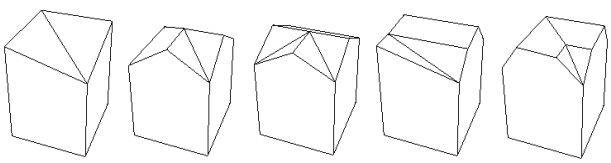


Figure 9. Examples of connecting roof shapes.

In summary, we determine a cell's roof type by comparing the points' normal vectors with the roof faces of all possible shapes and compute the percentage of points that fit the direction of the roof part they are inside. For a gabled roof, e.g., we divide the cell into two equal parts, distribute the points accordingly and count the number of points whose normal vectors are in accordance with the respective side (see Figure 10). Each roof type defines one or more parts, whose size may or may not be dependent on the roof parameters. E.g., the ridge line length of a hipped roof is variable and therefore affects the size of the four roof parts. The longer the ridge line grows, the smaller the two side hips become. This affects how accurately the shape can be determined.

Figure 10. The face normal directions of the four basic roof shapes: flat, shed, gabled and hipped. The flat roof face shows upwards.

2.2.1 Flat, Shed and Gabled Roof: When considering all junction elements, these basic shapes make up over twenty different shapes. The high number comes from the fact, that non-symmetric shapes can be rotated four times, resulting each time in a new shape. Only rotational symmetric shapes result in one shape and axial symmetric shapes in two shapes.

To efficiently determine if the points fit any of these basic roof types, or a connecting shape thereof, each cells footprint is broken into eight sections. For each section, the points are classified as pointing up, north, east, south and west depending on the cells orientation, where the first side of a cell is considered the south side. For a point to be classified as up, the angle between the point's normal direction and the upward vector must be below 30 degrees. For the other four classes, the 2D component of the point's normal vector must point more towards that side than to the other three, which reflects an angle below 45 degrees. Once all the points are classified, the percentage of matching points can be simply added up for all shapes.

Figure 11 shows four types of gabled roofs. For these classes of roof shapes, also the corner elements are used as they are basically free to compute. The basic gabled shape is axial symmetric and therefore only has two variants, the corner- and T-junctions can be rotated four times and therefore result in four variants each and the cross-junction is axial symmetric and therefore has one variant. The number of matching points for the gabled roof can be easily computed by adding the number of points in the green sections that show northwards and the number of points in the red sections that show southwards. The other shapes are computed accordingly, where the points in the blue sections must show westwards and the points in the yellow sections eastwards.

Once the points have been distributed to the eight sections and classified according to their normal direction, the time to do the summation is neglectable. This makes roof shapes whose shape can be reduced to the eight sections very appealing.

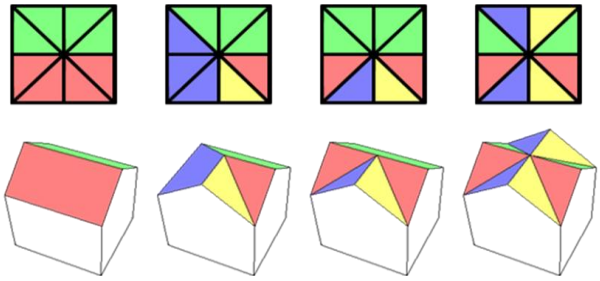


Figure 11. Gabled roof and its corner-, T- and cross-junctions and the direction points inside a particular face must show to.

2.2.2 Hipped Roof: For hipped (and other roof shapes that cannot be as easily divided into the eight sections as the aforementioned shapes), the roof area is divided individually. This is, however, not as efficient as before and some assumptions have to be made for some shapes. E.g. the ridge length of a hipped roof should be variable, but we assume that all four slopes are the same, which enforces a certain ridge length. This way only one variant must be evaluated, but it still reliably differentiates a hipped from e.g. a tent or gabled roof.

2.2.3 Berliner Roof: The Berliner roof is an asymmetric roof shape, which is basically a shed roof disinclined slightly to the back side. By having a steep slant at the front and sometimes also at the back side, the roof appears to be gabled from a pedestrians point of view. This shape is very common for Berlin apartment houses build during the period of promoterism in the 19th century.

To identify the front side of a cell with a possible Berliner roof, we seek the side closest to the building's oriented bounding rectangle. If the cell is a corner cell, or if all cells are side by side, then two or more sides of the cell should be within closest distance to the bounding rectangle. Here, the side with the highest number of normal vectors pointing towards to is determined. This is in most cases the back side. Both methods are necessary, as the second one generally fails more often, but is the only one that works for the latter case.

Then, the distances from the front and back side to the two fake ridge lines are determined using a plane sweep approach. At the front ridge line, the 2D components of the points' normal vectors show in opposite directions. As for the back ridge line, we say that all points' normal vectors with an angle below 30 degree compared to the upward vector belong to the shed part of the roof. Using these two criteria, we can accurately determine the two ridge lines that separate the three roof regions. Their height is computed from the plane equations estimated from the points of the two steep slant sections.

2.3 Parameter Estimation

Roof parameters vary from shape to shape. However, all shapes have one eaves height and up to two ridge heights, which are to be estimated from the LIDAR points. Among others, the cell's footprint defines the directions of the eaves and ridge lines. As all face slopes are linearly related, it allows determining them at once by simply estimating one plane equation from the given points. While one face defines a reference system, the points in other faces are translated into it accordingly. From the resulting plane equation, the eaves and ridge heights can be determined from the reference face. The resulting shape parameters best fits all the faces to the input points.

2.4 Roof Junctions:

Cells that have neighbor cells at two consecutive sides or at three or more sides are examined again. These cells are candidates for connecting shapes. Based on the shape types, the parameters and the arrangement of the neighbor cells, compatible connecting shapes are determined. The one that connects the most neighbor cells to a sound roof structure is then chosen and its parameters determined from the parameters of the neighbor cells.

2.5 Manuel Editing

Because not all roof structures can be fully automatically reconstructed, there is a need for manual editing. In our editing tool, the decomposition lines can be copied, added, deleted, translated and rotated. The cells' roof shapes are automatically reconstructed after every manual step, so that the operator can immediately see the results. Once the cell decomposition fits the roof's shape, the cell parameters can be manually adjusted or even copied from other cells. If the decomposition produces too many small cells, then their number can be decreased by a merging operation.

Even though editing the building models using decomposition lines is not so straight-forward, we found that operators got used to it very quickly and can efficiently produce even landmarks with complex geometry. The manual mode also allows for more complex roof shapes like mansard, cupola, barrel and even some detail elements like dormers.

3. PROJECTS

While still in development, we started using the reconstruction software in a real production environment. Several large-area projects have since been successfully completed. The feedback in the early stages of development helped us to recognize and adapt to arising problems. Two of our early projects were the 3D reconstruction of East Berlin and Cologne, two major cities in Germany. The 3D city model of Berlin is also available online for use in Google Earth (Berlin 3D, 2009).

3.1 East Berlin, Germany

The first project with our new software was to perform a 3D building reconstruction from Berlin's LIDAR data. The total area of the project was 498 km² with approximately 244,000 buildings. The project was an extension of the original 3D City model of Berlin, Germany, which is to date still the largest city model transported to the Google Earth platform. Input data included a DTM, airborne LIDAR and building footprints. See Figure 12 and Figure 13 for the resulting model.

Due to the large number of buildings in East Berlin and project time constraints, photogrammetric extraction was immediately deemed as being too time consuming and costly. It was therefore decided to use LIDAR data instead. All LOD 2 building models are geo-referenced geometry, which were later textured using aerial oblique imagery.

The Berliner Roof– a particularly unusual roof type typically found on many buildings in Berlin – presented a challenge as well as numerous inner courtyards presented problems during extraction. Therefore, the reconstruction approach had to be adapted to automatically detect this unique roof structure. As a result, a total of 17 individual roof types have been additionally integrated into the software to enable greater accuracy during reconstruction and to reduce the amount of manual editing needed.

As the software was constantly improved during the duration of the project, the amount of manual editing needed for the reconstructed 3D buildings was reduced from 30 percent in denser areas to 20 percent; manual editing for the outer lying areas also experienced a sharp improvement: from 20 percent to 15 percent.

Figure 12. 3D city model of East Berlin.

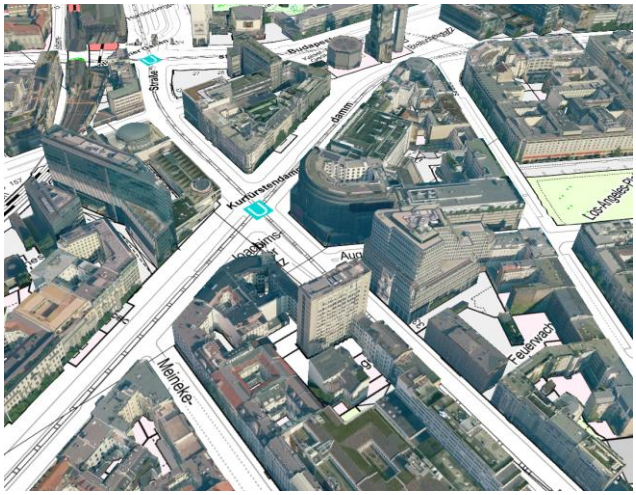


Figure 13. 3D city model of Berlin textured from oblique images showing part of the prominent Kurfürstendamm.

3.2 Cologne, Germany

The existing 3D city model of Cologne is used by several administrative departments as a complement to the existing GIS data inventory held by the Cologne Survey Department. It was created from the basis of building storeys using two-dimensional footprints; therefore the true heights of the buildings were not accurate. In order to produce a more realistic representation of Cologne in 3D to be used for urban planning and emergency response, the survey department decided to use the data from the most recent LIDAR flyover to perform a real 3D building reconstruction. See Figure 14 and Figure 15 for the results.

Cologne's city boundaries encompass approximately 415 km² with 280,000 buildings; therefore it was decided to use airborne

LIDAR instead of photogrammetry. In many areas of the inner city, Cologne has an extreme building density, which complicated a clean separation of building geometry and roof forms, even though the building outlines contained in the ground cadastre map were examined beforehand for their accuracy.

In addition, there were many special building structures such as churches that had to be extracted from the airborne LiDAR data. Pre-processing efforts were further complicated by the fact that Cologne's ground plan data was outdated or incomplete as several new buildings that had been erected and still others had been torn down since the last update made to the ground cadastre map.

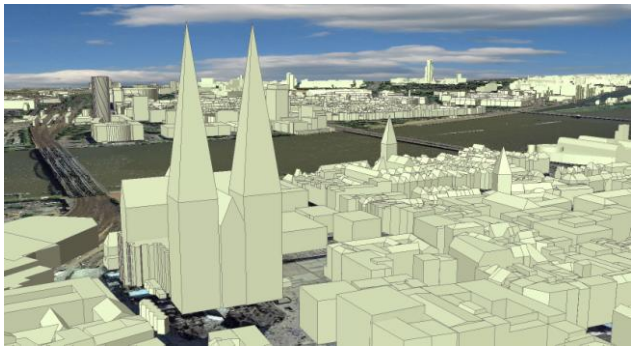


Figure 14. 3D city model of Cologne.

After a careful study of the digital ground map it was determined that first several adjustments had to be made, for example removing underground buildings and structures such as parking garages and identify torn down buildings. This required examining the discrepancies between the DTM, DSM and building outlines to create the final 3D city model.

Finally, many larger buildings appeared in several different attribute tables containing sometimes conflicting information, therefore presented a challenge for both the client as well as the operators because these buildings still needed to be reconstructed without altering their original building footprints.

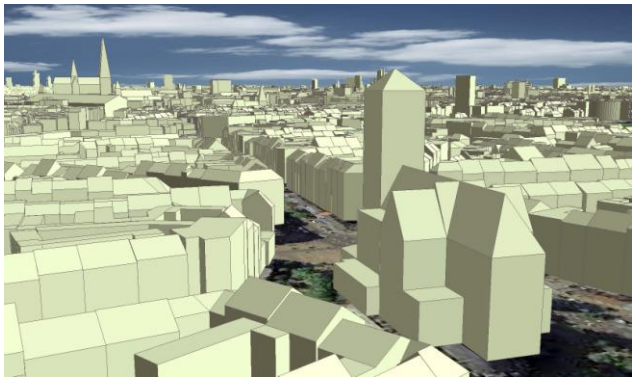


Figure 15. 3D city model of Cologne.

We have completed a wide-area 3D city model in LOD2 for the whole of Cologne by the end of September 2008. This new model will be integrated into the existing model, thereby replacing the GIS data with a much more accurate representation of the building heights. Because the inner city buildings will receive realistic façade textures, highly accurate building heights and roof structures as well as building details were a key project requirement.

The entire model will be used a decision making tool for urban planning and serves as a visualisation tool and complement to Cologne's Master Plan. The amount of overall manual post editing required with the software has been reduced since working on the East Berlin model to 15 percent.

4. CONCLUSION AND FUTURE WORK

We have presented an approach for the automatic reconstruction of 3D building models from LIDAR data and existing ground plans. It is based on an algorithm to decompose given footprints into sets of nonintersecting cells, for which roof shapes are then determined from the normal directions of the LIDAR points. The validity of this approach has been proven effective, as can be judged by the 3D city models of East Berlin and Cologne.

The next step is to increase the amount of detail by loosening some of the restrictions of our shapes and by making them more flexible. This is already possible in manual editing. However, to increase both the richness in detail and the automation, we plan to integrate a segmentation of the roof points to selectively decompose the footprints without generating more cells.

5. REFERENCES

Arefi, H., Engels, J., Hahn, M. and Mayer, H., 2008. Levels of Detail in 3D Building Reconstruction from LIDAR Data. In: *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences, Vol. XXXVII-B3b*.

Berlin 3D, 2009. 3D-Stadtmodell Berlin. http://www.3d-stadtmodell-berlin.de (accessed 6 April 2009)

Brenner. C., 2005. Building Reconstruction from Images and Laser Scanning. In: *International Journal of Applied Earth Observation and Geoinformation (Theme Issue 'Data Quality in Earth Observation Techniques), Vol. 6 (3-4)*, p. 187-198.

Kada, M. 2007. Scale-Dependent Simplification of 3D Building Models Based on Cell Decomposition and Primitive Instancing. In: Spatial Information Theory: 8th International Conference, COSIT 2007, pp. 222-237.

Kolbe, T.H., 2009. Representing and Exchanging 3D City Models with CityGML. In: *Lee, Zlatanova (Eds.): 3D Geo-Information Sciences*, Springer-Verlag Berlin Heidelberg, pp. 15-32.

Möser, S., Wahl, R. and Klein, R., 2009. Out-Of-Core Topologically Constrained Simplification for City Modeling from Digital Surface Models. In: *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences, Vol. XXXVIII-5/W1*.

Sohn, G., Huang, X. and Tao, V., 2008. Using a Binary Space Partitioning Tree for Reconstructing Polyhedral Building Models from Airborne Lidar Data. In: *Photogrammetric Engineering & Remote Sensing, Vol. 74 No. 11*, pp. 1425-1438.

A SEMI-AUTOMATIC APPROACH TO OBJECT EXTRACTION FROM A COMBINATION OF IMAGE AND LASER DATA

S. A. Mumtaz [a, *], K. Mooney [a]

[a] Dept. of Spatial Information Sciences, The Dublin Institute of Technology, Bolton Street, Dublin 1, Ireland (salman.mumtaz, kevin.mooney)@dit.ie

Commission III, WG III/4

KEY WORDS: LiDAR, Object Extraction, Data fusion, Buildings, Trees, Roads

ABSTRACT:

The aim of the authors' research is to develop an automated or semi-automated workflow for the extraction of objects such as buildings, trees and roads for noise mapping and road safety purposes. The workflow must utilise national airborne spatial data available throughout the country and be capable of robust incorporation in the noise modelling systems of a national roads authority. This paper focuses on the extraction of multiple objects by fusing data captured by two independent sensors, namely the Leica ADS40 aerial camera and the Leica ALS50 airborne laser scanner (LiDAR). A workflow has been developed for the extraction of objects utilizing height values from a normalised DSM generated using LiDAR or aerial images, multiple LiDAR echo data and NDVI (Normalized Difference Vegetation Index) data computed from multispectral ADS40 data.
Major tasks include LiDAR data classification, segmentation and its integration with the information extracted from aerial images. Buildings are extracted first and this facilitates the extraction of other objects. Preliminary results of this semi-automated process indicate high completeness rates for buildings trees and roads but 60% quality rates (e.g. buildings). Quality may be improved by manual extraction of small objects but continuing research is focussed on reducing reliance on such manual intervention.

1. INTRODUCTION

The National Roads Authority (NRA) in Ireland is responsible for generating noise maps in the environment of roads used by more than 8220 vehicles per day. According to the EU noise directive this exercise must be repeated every five years. Inputs for generating the noise maps include terrain model, location and dimension of buildings, trees, noise barriers and the geometric properties of roads. Capturing this data using field surveys or digital images is time consuming and expensive, especially if the same exercise must be repeated every five years. It is the intention of this work that all required objects be extracted using automatic or semiautomatic techniques from LiDAR and aerial image data of the type available from the National Mapping Agency of Ireland, OSi (Ordnance Survey of Ireland). Later the extracted information can be easily combined and analyzed along with noise data in a GIS system. For noise mapping, building detail or tree models are not required. Buildings or trees boundaries with height information are sufficient.

High resolution image and LiDAR sensors (ADS40 & ALS50) were used to capture the data for a part of County Sligo in the northwest of Ireland. Digital images were captured in April 2007 with a ground resolution of 15 cm. LiDAR data were captured separately in May 2007 at a flying height of 1241 m with a swath width of 800 m, resulting in an average point density of approximately 2 points/m^2. The ALS50 sensor recorded position, multiple echoes and intensity of the returning pulse.

The area selected for processing is about 3 km^2 and is covered by a single image strip and four LiDAR strips. This eliminates the necessity for bundle block adjustment and ground control point acquisition. The reason for relying completely on direct geo-referencing in this research is the fact that in many situations ground control points may not be available. Strip adjustment of the LiDAR data was performed using the Terra Match application from TerraSolid.

1.1 Motivation

In recent years, research on automated object extraction has increased because of the increased use of GIS (Geographical Information Systems) with the consequential need for data acquisition and update.

Digital Photogrammetry is considered to be one of the most precise methods for capturing large scale data for GIS analysis from high resolution aerial images. However, it requires significant resources to digitize all objects of interest. As detailed high resolution digital images are regularly acquired as part of the national programme of OSi, it is considered important to develop automatic or semi-automatic techniques to exploit their potential for applications such as noise modelling involving the extraction of objects such as buildings, trees and roads.

LiDAR can provide high density 3D point clouds in a very short time with acceptable horizontal and high vertical accuracy. OSi also acquires national LiDAR data using the ALS50 sensor from Leica Geosystems. The availability to the national roads authority of Ireland of both of these high resolution data sources provides the impetus for this research.

However, the development in sensor technology is far more rapid than the advancements in automatic or semi automatic object extraction. Moreover there is still a large gap between

* Salman Ali Mumtaz

the theoretical work on fully automated object extraction and practical applications of the same (Mayer 2008). Success in automatic object extraction will also help in determining changes that occur between noise surveys (5 years) by comparing the extracted objects at the different epochs and should speed up the updating process of the GIS database.

1.2 Related Work

LiDAR has been extensively used for the generation of both DSMs (Digital Surface Model) and DTMs (Digital Terrain Model). Different classification methods have been used for the classification of terrain and off terrain points (Sithole & Vosselman, 2003). Different approaches have been used for the detection and reconstruction of buildings from LiDAR data (Brunn & Weidner, 1997 and Clode et al., 2004). Haitao et al. (2007) used aerial images and LiDAR data for land cover classification based on SVM (Scalable Vector Machine). Haala & Brener (1999) also used the combination of multispectral imagery and LiDAR data for the extraction of buildings, trees and grass covered areas. Trees and grass covered areas were classified easily from the multispectral imagery but were found difficult to separate. Similarly, trees and buildings were separated using height differences between DSM and DTM. Both data sources were combined in order to identify the three classification types. Rottensteiner et al. (2004) classified land cover into four different classes namely, buildings, trees, grass lands, and bare soil. This was achieved by combining LiDAR data and multispectral images. Prior to performing building detection by data fusion based on the theory of Dempster-Shafer, the LiDAR data was pre-processed to generate a DTM. For the extraction of roads different information sources such as multispectral images from airborne and space borne sensors were used. Clode et al. (2007) used only LiDAR for road extraction. Despite encouraging results, there are still many fundamental questions to be answered for road extraction in urban areas (Mayer et al., 2008).

2. METHOD

The method under investigation is based on a workflow that identifies and classifies buildings, trees and other objects by fusing the information from LiDAR and aerial image data. This information includes the normalised digital surface model (NDSM) and multiple echoes from the LiDAR data together with Normalized Difference Vegetation Index (NDVI) data generated from the airborne imagery. The method is depicted in Figure 1. Three major task groups may be identified, namely the image group, LiDAR group and object extraction group.

3. WORKFLOW

Within the image group of tasks, the first step is to produce orthophotos for each spectral channel of the ADS40 sensor, i.e. Red (R), Green (G), Blue (B) & Near Infrared (NIR). For these orthophotos, the required DSM can be created relatively automatically using the panchromatic forward and backward image data captured by the ADS40 sensor. The effect of DSM quality on orthophoto generation is shown in Figure 2. The upper part of the figure shows a rectified building using a DSM generated by aerial images and the lower part shows the same building rectified using a DSM from LiDAR data.

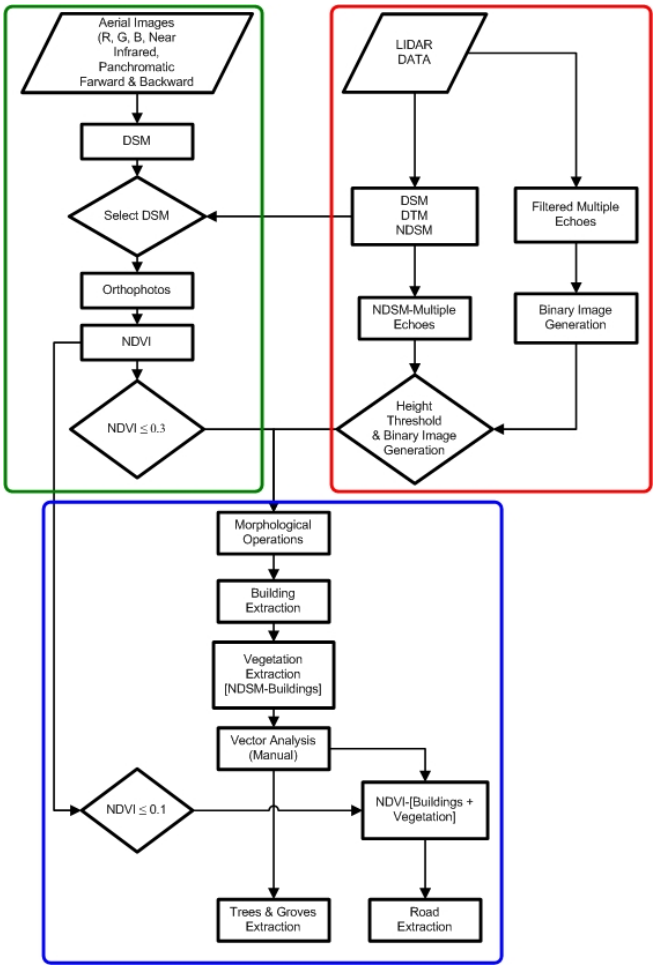


Figure 1: Method Workflow

Figure 2: Effect of DSM on Building Rectification

In case of the building illustrated in Figure 2, a DSM created from LiDAR data with a resolution of 0.5 m (the lower example) provided sharper edges compared to that generated from the image DSM and was used in the generation of the orthophotos. As a prerequisite to this step the quality of the registration between the airborne imagery and the LiDAR data must be verified. The Nearest Neighbourhood method was used as a sampling method for orthophoto generation. Separate

orthophotos were generated from the R, G, B and NIR channels (a true colour orthophoto was also generated) and the NDVI was calculated using the following formula:

$$NDVI = (NIR-R) / (NIR+R)$$

NDVI values range from -1 to +1 which suggests that if the pixel value is close to -1 it does not belong to healthy vegetation or vice versa. As a result, NDVI data could assist in separating vegetation from buildings in a DSM.

Figure 3 shows the effect of sun position on selecting an NDVI threshold to separate buildings from vegetation. In the shadow area NDVI values are larger than the portion of the building directly facing the sun. A larger threshold value of 0.3 was selected to differentiate between buildings and vegetation. Because of this large threshold value some vegetation also appears with the buildings.

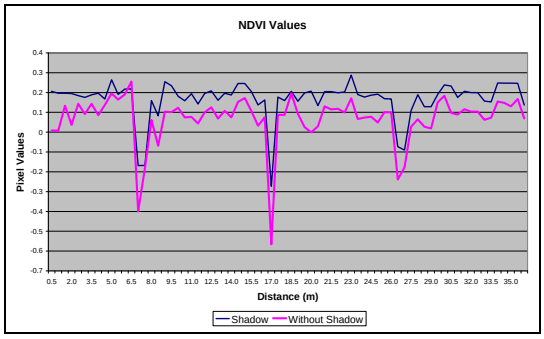


Figure 3: NDVI Threshold for Buildings

Within the LiDAR group of tasks, the first step is the generation of a DSM and DTM from the LiDAR data (TerraSolid software was used). In order to get the absolute height of the objects the DTM was subtracted from the DSM to give the NDSM. A further refinement of the NDSM can then be achieved by making use of multiple LiDAR echo data. These occur from building edges and trees. Figure 4 (Clode et al., 2005) shows how the laser beam interacts with building edges and trees.

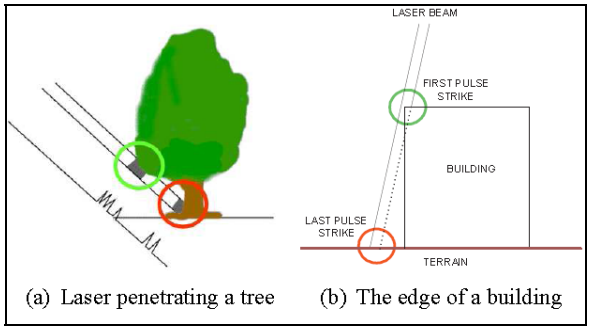


Figure 4: Laser Interactions

Firstly, the filtered multiple echoes (figure 5) were converted into an image. Gaps between pixels of less than 3 metres were filled and a binary image was generated. Selecting a value for gap filling depends on the density of the original point data. If the density is high a small value can serve the purpose but it should not be too high that it causes individual trees close to each other to merge.

The separation of multiple echo data (Figure 5) from the NDSM, by multiplication by the binary data, results in data only from those objects that record a single reflection. These include buildings and other solid objects but also vegetation that returned single echoes.

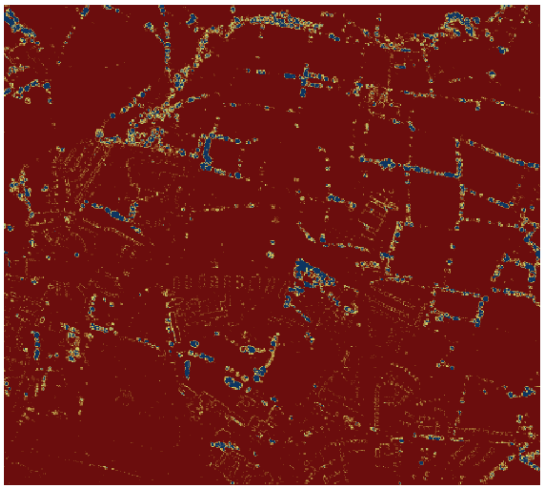


Figure 5: Filtered Multiple Echoes

The next step was to apply a height threshold of 2.5m to the NDSM to eliminate objects such as hedges, cars etc and the resultant NDSM containing buildings, vegetation and other tall objects was converted to a binary image. All pixels having a value lower than or equal to 2.5 m were assigned a zero value and the remainder a value of one (Figure 6).

A morphological operation such as closing and opening was used for filling small gaps in the binary image. Care should be taken as too many repetitions can result in rounding of the sharp building edges and loss of important detail.

This binary image contains pixels that belong to buildings and remaining trees and needs further classification. This was achieved by introducing the NDVI image described as part of the image group of tasks.

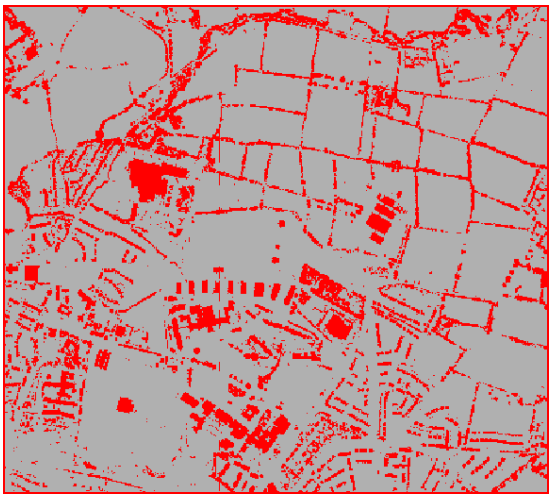


Figure 6: Binary Image with Height Threshold

The NDSM and the NDVI images were combined and the maximum likelihood classification method was used for the

extraction of buildings (Figure 7). The area was calculated for each building after conversion to vector format and used as a threshold for separating main buildings from smaller structures in front or behind the main buildings. These areas also include trucks and vans on the roads or parked near to buildings and are detected as buildings. These areas were retained and classified later into buildings, vegetation or other objects.

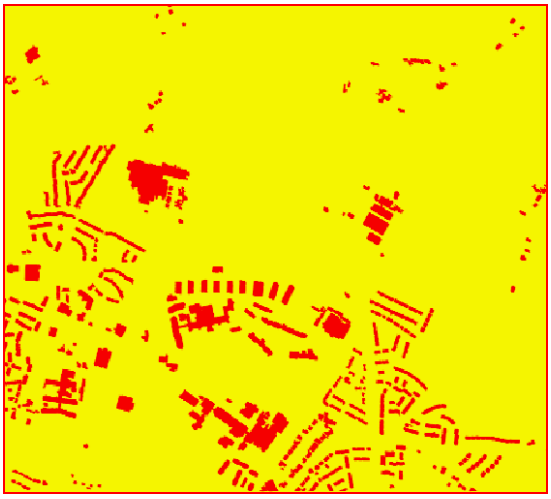


Figure 7: Extracted Buildings

To assign each building an individual height, building centroids were determined. For each centroid, heights were determined from the DSM by bilinear interpolation and the same process was used for determining vegetation height.

For the purpose of vegetation extraction, the final building data was subtracted from the NDSM. This resulted in vegetation present in the NDSM layer that is higher than 2.5m. However, the filtered multiple echo data (Figure 5) was also processed further. First, intermediate and last echoes represent reflections from the edges of buildings and trees. Once buildings were classified, building boundaries were used as an input to remove all multiple echo points that belong to building edges. Multiple reflections from large trees, together with compactness (area/perimeter^2) were used to classify large single trees and groves (Figure 8).

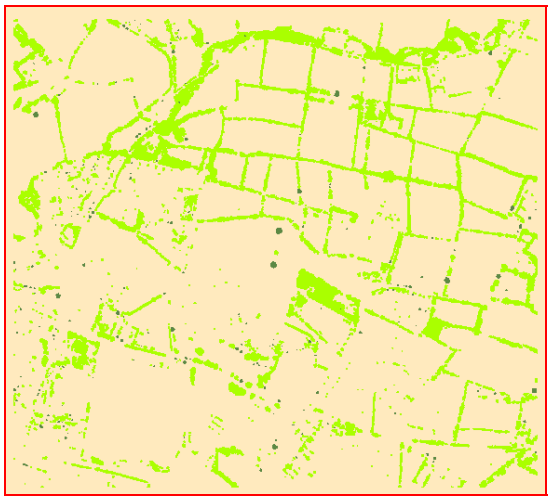


Figure 8: Extracted Trees and Groves

The remaining objects to be classified were roads which are part of the generated DTM. NDVI data below a threshold of 0.1 and the previously classified objects were used. Using the threshold eliminates most of the area having vegetation but does not help much in the areas with barren land. Their spectral signature value is also very close to the roads. Even for the roads the reflection value is not constant. It varies with age and type of material used in the road surface. Previously extracted buildings and vegetation were subtracted from the NDVI to extract road candidates.

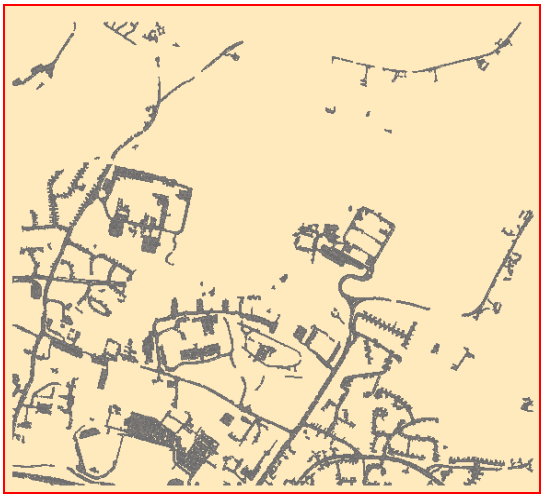


Figure 9: Extracted Roads

Gaps are present in the extracted roads and this is because of the trees and building shadows. This section of the road is not visible in aerial photos and LiDAR and needs further research for its successful extraction. Finally all the extracted objects were combined together and integrated into a specialized GIS system (Figures 10 and 11).

4. CLASSIFICATION ASSESSMENT

Results of the workflow were evaluated after the method of Heipke et al. (1997). Using this method, three different states for any feature can be identified (Hatger, 2006).

True positive (TP) - A phenomenon that is present within the input data and that has successfully been identified within the output data.

False positive (FP) - A phenomenon that is not present within the input data but that has falsely been found to be a phenomenon by the algorithm. Thus it is written to output data.

False negative (FN) - A phenomenon that is present within the input data but that has not been identified by the algorithm and therefore has been omitted from the output data.

Then we define Completeness, Correctness and Quality by

$$\text{Completeness} = TP / (TP + FN)$$

$$\text{Correctness} = TP/ (TP + FP)$$

$$\text{Quality} = TP / (TP + FP + FN)$$

Object	Completeness (%)	Correctness (%)	Quality (%)
Buildings	98	72	60
Trees	99	83	80
Roads	78	72	62

Table 1: Accuracy Assessment

Figure 10: Extracted Objects & Orthophoto

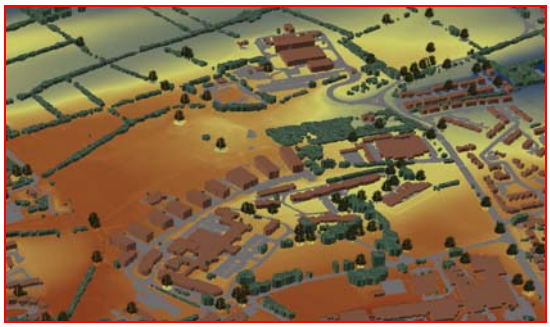


Figure 11: 3D Model (County Sligo)

5. DISCUSSION

The classification results, indicated in Table 1, are the result of automated processes, depending on the choice of appropriate parameters and thresholds.

Building extraction is the first step in the classification process and is therefore important for the extraction of further objects. A completeness value of 98% implies that the adopted strategy has been successful in the identification of these objects. However correctness and quality values are significantly less than completeness due to the influence of FP values. FP values in buildings arise from large trucks or industrial installations incorrectly identified as buildings. However, FN values (i.e. missed buildings) mostly occurred for small buildings less than 50 sq. m.

In a subsequent manual step, these small buildings were individually identified and included in the final building layer. The identification of all buildings in the area in this way allowed the extracted vegetation data to be improved, which later helped in the extraction of roads. Commercial or residential buildings, having glass roofs or green colour were missed in the NDVI layer but they existed in the NDSM and were added to the building layer manually. Many small sheds were identified in the backyards of houses which are not part of the buildings, which significantly reduced the correctness value.

Vegetation was extracted by subtracting the building layer from the NDSM. Very small buildings which appeared in the vegetation layer were identified and manually added to the building layer. Continuing research is targeted at reducing the dependence on such manual steps.

Multiple reflections, size and compactness were used to separate single trees from groves. However, the LiDAR sensor can efficiently differentiate between multiple reflections only where their height differences are significant.

Roads appear to be the most difficult objects to extract. They are part of the DTM and have spectral reflectance, which varies a lot in a single image. Setting a NDVI threshold helps identify the areas where there is vegetation or not. Reflections from barren land or walking trails in the fields also have very low NDVI values. Roads which are not covered by building shadows or trees are detected successfully. Road markings of different colours also affect the extraction process. Roads connecting houses to the road are of different materials and need to be classified separately.

6. CONCLUSION

The accuracy of the generated orthophoto is critical for any classification technique using LiDAR and aerial images. Due to the nature of the push broom sensor and the configuration of the test flight (no overlap along strip and 15% overlap between strips) there is no possibility to combat limitations in the Red, and NIR orthoimages. Occluded areas and ghosting of building roofs (in the across flight direction) cannot be corrected adequately and the building roof structure is completely damaged in the areas close to strip edges. This is a major disadvantage in the identification and modelling of building roof structures. Ground control points, where available, should be used for the verification of the registration of the LiDAR point cloud and aerial images. In this approach we relied completely on orientation from GPS\INS data but for future research ground control points will be acquired and the accuracy of the image registration will be measured.

For the purpose of the NDVI image, the Red and NIR channels exhibit excessive tree pixels in the extremities due to tree lean and the structure of the resulting NDVI image will therefore not match the structure of the DSM. This requires further investigation. The DSM quality can also be improved by incorporating building foot prints if available.

LiDAR and aerial images should, ideally, not be captured separately. Objects which exist in the images might not exist in the LiDAR data and it is time consuming to identify and separate those points, especially vehicles on roads or in parking areas. In addition, if the time delay is significant, the vegetation may change considerably.

Results from the automatic and semi-automatic stages of this workflow are encouraging. Limitations identified above are the subject of continuing research.

7. REFERENCES

Clode, S. and Rottensteiner, F. 2005. Classification of Trees and Powerlines from Medium Resolution Airborne Laser Scanner Data in Urban Environments. Proceedings of Workshop on Digital Image Computing, Brisbane, Australia, pp. 97-102.

Clode, S., Rottensteiner, F., Kootsookos, P. and Zelniker, E., 2007. Detection and Vectorization of Roads from LiDAR Data. *Photogrammetric Engineering and Remote Sensing*, 73(5), pp. 517-535.

Rottensteiner, F. Trinder, J. Clode, S. Kubik, K. Lovell, B. 2004. Using the DempsterShafer Method for the Fusion of LiDAR Data and Multi-spectral Images for Building Detection. Proceedings of the 17th International Conference on Pattern Recognition, Vol. 2, pp. 339 – 342.

Fugro International. FLI-MAP 400 Specifications. [Online]. Available at: http://www.flimap.com/site47.php (accessed: 26th June 2009).

Haitao, L. Haiyan, G., Yanshun, H. and Jinghui, H., 2007. Fusion of High Resolution Aerial Imagery and LiDAR Data for Object-Oriented Urban Land-Cover Classification Based on SVM. ISPRS Workshop on Updating Geo-spatial Databases with Imagery & The 5th ISPRS Workshop on DMGISs, Urumchi, Xingjiang, China. [Online]. Available at: www.commission4.isprs.org/urumchi/papers/179-184%20Haitao%20Li.pdf (accessed: 26th June 2009).

Hatger, C., 2005. On the Use of Airborne Laser Scanning Dat to Verify and Enrich Road Network Features , Proceedings of ISPRS Technical Commission III Symposium , Enschede, Netherlands.

Heipke, C., Mayr, H. , Wiedemann, C. and Jame, O., 1997. Evaluation of Automatic Road Extraction. *International Archives of Photogrammetry and Remote Sensing*, XXXII (3/2W3), 56.

Leica Geosystems Incorporation. [Online]. Available at: http://www.leica-geosystems.com/corporate/en/lgs_57627.htm (accessed: 12th October 2008).

.

Mayer, H., 2008. Object Extraction in Photogrammetric Computer Vision. *ISPRS Journal of Photogrammetry and Remote Sensing*, Volume 63, Issue 2, pp. 213-222.

Mayer, H., Hinz, S. and Stilla, U. 2008. Automated Extraction of Roads, Buildings and Vegetation from Multi-source Data, *Advances in Photogrammetry, Remote Sensing and Spatial Information Sciences:* ISPRS Congress Book.

N. Haala and C. Brenner, 1999. Extraction of Buildings and Trees in Urban Environments. *ISPRS Journal of Photogrammetry and Remote Sensing*, Vol. 54, pp. 130–137.

Sithole, G and Vosselman, G., 2003. Comparison of Filtering Algorithms. Proceedings of the ISPRS working group III/3 workshop on 3-D reconstruction from airborne laserscanner and InSAR data, Dresden, Germany, 8-10 October .

COMPLEX SCENE ANALYSIS IN URBAN AREAS BASED ON AN ENSEMBLE CLUSTERING METHOD APPLIED ON LIDAR DATA

P. Ramzi*, F. Samadzadegan

Dept. of Geomatics Engineering, Faculty of Engineering, University of Tehran, Tehran, Iran - (samadz, pramzi)@ut.ac.ir

Commission III, WG III/4

KEY WORDS: LIDAR, Feature, Object, Extraction, Training, Fusion, Urban, Building

ABSTRACT:

3D object extraction is one of the main interests and has lots of applications in photogrammetry and computer vision. In recent years, airborne laser-scanning has been accepted as an effective 3D data collection technique for extracting spatial object models such as digital terrain models (DTM) and building models. Data clustering, also known as unsupervised learning is one of the key techniques in object extraction and is used to understand structure of unlabeled data. Classical clustering methods such as k-means attempt to subdivide a data set into subsets or clusters. A large number of recent researches have attempted to improve the performance of clustering. In this paper, the boost-clustering algorithm which is a novel clustering methodology that exploits the general principles of boosting is implemented and evaluated on features extracted from LiDAR data. This method is a multi-clustering technique in which At each iteration, a new training set is created using weighted random sampling from the original dataset and a simple clustering algorithm such as k-means is applied to provide a new data partitioning. The final clustering solution is produced by aggregating the weighted multiple clustering results. This clustering methodology is used for the analysis of complex scenes in urban areas by extracting three different object classes of buildings, trees and ground, using LiDAR datasets. Experimental results indicate that boost clustering using k-means as its underlying training method provides improved performance and accuracy comparing to simple k-means algorithm.

1. INTRODUCTION

Airborne laser scanning also known as LiDAR has proven to be a suitable technique for collecting 3D information of the ground surface. The high density and accuracy of these surface points have encouraged research in processing and analyzing the data to develop automated processes for feature extraction, DEM generation, object recognition and object reconstruction. In LiDAR systems, data is collected strip wise and usually in four bands of first and last pulse range and intensity (Arefi et al, 2004). Clustering is a method of object extraction and its goal is to reduce the amount of data by categorizing or grouping similar data items together. It is known as an instance of unsupervised learning (Dulyakarn and Rangsanseri, 2001). The grouping of the patterns is accomplished through clustering by defining and quantifying similarities between the individual data points or patterns. The patterns that are similar to the highest extent are assigned to the same cluster. Generally, clustering algorithms can be categorized into iterative square-error partitional clustering, hierarchical clustering, grid-based clustering and density-based clustering (Pedrycz, 1997; Jain et al., 2000).

The most well-known partitioning algorithm is the k-means which is a partitional clustering method so that the data set is partitioned into k subsets in a manner that all points in a given subset are closest to the same center. In other words, it randomly selects k of the instances to represent the clusters. Based on the selected attributes, all remaining instances are assigned to their closer center. K-means then computes the new centers by taking the mean of all data points belonging to the same cluster. The operation is iterated until there is no change in the gravity centers. If k cannot be known ahead of time, various values of k can be evaluated until the most suitable one is found. The effectiveness of this method as well as of others relies heavily on the objective function used in measuring the distance between instances. The difficulty is in finding a distance measure that works well with all types of data (Jane and Dubes, 1995). Some attempts have been carried out to improve the performance of the k-means algorithm such as using the Mahalanobis distance to detect hyper-ellipsoidal shaped clusters or using a fuzzy criterion function resulting in a fuzzy c-means algorithm (Bezdek and Pal, 1992). A few authors have provided methods using the idea of boosting in clustering (Frossyniotis et al., 2004; Saffari and Bischof, 2007; Liu et al., 2008).

1.1 Related Work

Boosting is a general and provably effective method which attempts to boost the accuracy of any given learning algorithm by combining rough and moderately inaccurate classifiers (Freund and Schapire, 1999). The difficulty of using boosting in clustering is that in the classification case it is straightforward whether a basic classifier performs well with respect to a training point, while in the clustering case this task is difficult since there is a lack of knowledge concerning the label of the cluster to which a training point actually belongs (Frossyniotis et al., 2004). The authors in (Frossyniotis et al., 2004) used the same concept, by using two different performance measures for assessing the clustering quality. They incorporated a very similar approach used in the original Discrete AdaBoost

* Corresponding author.

(Freund and Schapire, 1996) for updating the weights and compared the performance of k-means and fuzzy c-means to their boosted versions, and showed better clustering results on a variety of datasets. (Saffari and Bischof, 2007) provided a boosting-based clustering algorithm which builds forward stage-wise additive models for data partitioning and claimed this algorithm overcomes some problems of Frossyniotis et al algorithm (Frossyniotis et al., 2004). It should be noted that the boost-clustering algorithm does not make any assumption about the underlying clustering algorithm, and so is applicable to any clustering algorithm.

However, most of the above methods are provided and evaluated on artificial or standard datasets with small sizes and the significance of improvement in object extraction using this method is not evaluated in urban areas. In this paper, the boost-clustering method is implemented and evaluated on two subsets of LiDAR data in an urban area. The results are then provided in the form of error matrix and some quality analysis factors used for the analysis of classification performance, and compared to the results of the core algorithm in boosting, simple k-means.

2. BOOSTING ALGORITHM

Boosting is a general method for improving the classification accuracy of any classification algorithm. The original idea of boosting was introduced by (Kearns and Valiant, 1998). Boosting directly converts a weak learning model, which performs just slightly better than randomly guessing, into a strong learning model that can be arbitrarily accurate. In boosting, after each weak learning iteration, misclassified training samples are adaptively given high weights in the next iteration. This forces the next weak learner to focus more on the misclassified training data. Because of the good classification performance of AdaBoost, it is widely used in many computer vision problems and some promising results have been obtained (Li et al., 2004). A few attempts have been accomplished to bring the same idea to the clustering domain.

2.1 Boosting Clustering

Boost-clustering is an ensemble clustering approach that iteratively recycles the training examples providing multiple clusterings and resulting in a common partition (Frossyniotis et al., 2004). In ensemble approaches, any member of the ensemble of classifiers are trained sequentially to compensate the drawbacks of the previously trained models, usually using the concept of sample weights. It is sometimes considered as a classifier fusion method in decision level. At each iteration, a distribution over the training points is computed and a new training set is constructed using random sampling from the original dataset. Then a basic clustering algorithm is applied to partition the new training set. The final clustering solution is produced by aggregating the obtained partitions using weighted voting, where the weight of each partition is a measure of its quality (Frossyniotis et al., 2004). Another major advantage of boost clustering is that its performance is not influenced by the randomness of initialization or by the specific type of the basic clustering algorithm used. In addition, it has the great advantage of providing clustering solutions of arbitrary shape though using weak learning algorithms that provide spherical clusters, such as the k-means. It is because the basic clustering method (k-means) is parametric, while the boost-clustering method is nonparametric in the sense that the final partitioning is specified in terms of the membership degrees $h_{i,j}$ and not through the specification of some model parameters.

This fact gives the flexibility to define arbitrarily shaped data partitions (Frossyniotis et al., 2004).

The utilized algorithm is summarized below (Frossyniotis et al., 2004):

1. Input: Dataset $(x_1,...,x_N), x_i \in \Re^d$, number of clusters (C) and maximum number of Iterations (T), Initialize $w_i^1 = 1/N$
2. for t=1to T
   a. produce a bootstrap replicate of original dataset
   b. apply the k-means algorithm on dataset to produce the cluster hypothesis $H_i^t = (h_{i,1}^t, h_{i,2}^t,..., h_{i,C}^t)$ where $h_{i,1}$ is the membership of instance i to cluster j
   c. if t>1, renumber the cluster indices of H^t according to the results of previous iteration
   d. calculate the pseudo-loss

$$\varepsilon_t = \frac{1}{2}\sum_{i=1}^{N} w_i^t CQ_i^t \qquad (1)$$

   e. set $\beta = \frac{1-\varepsilon_t}{\varepsilon_t}$
   f. if $\varepsilon_t > 0.5$, go to step 3
   g. update distribution W:

$$W_i^{t+1} = \frac{w_i^t \beta_t^{CQ_i^t}}{Z_t} \qquad (2)$$

   h. compute the aggregate cluster hypothesis:

$$h_{ag}^t = \arg\max_{k=1,...,C} \sum_{\tau=1}^{t} \left[\frac{\log(\beta_\tau)}{\sum_{j=1}^{t}\log(\beta_j)} h_{i,k}^\tau \right] \qquad (3)$$

3. Output the final cluster hypothesis $H^f = H_{ag}^T$

In the above algorithm, a set X of N dimensional instances x_i, a basic clustering algorithm (k-means) and the desired number of clusters C are first assumed. At each iteration t, the clustering result will be denoted as H^t, while H_{ag}^T is the aggregate partitioning obtained using clustering of previous iteration. Consequently, at the final step, H^f is will be equal to H_{ag}^T. In this algorithm, at each iteration t, a weight w_i^t is computed for each instance x_i such that the higher the weight the more difficult is for x_i to be clustered. At each iteration t, first a dataset X^t is constructed by sampling from X using the distribution W^t and then a partitioning result H^t is produced using the basic clustering algorithm. In the above methodology an index CQ_i^t is used to evaluate the clustering quality of an instance x_i for the partition H^t. In our implementation, index CQ is computed using equation 4.

$$CQ_i^t = 1 - h_{i,good}^t - h_{i,bad}^t \qquad (4)$$

where

$h_{i,good}^t$ = the maximum membership degree of x_i to a cluster.

$h_{i,bad}^t$ = the minimum membership degree to a cluster.

Here, the membership degree $h_{i,j}$ for every instance x_i to cluster j, is produced based on the Euclidean distance d:

$$h_{i,j} = \frac{1}{\frac{d(x_i, \mu_j)}{\sum_{k=1}^{C} d(x_i, \mu_k)}} \quad (5)$$

where
$\mu_j \in \Re^d$ = cluster center.
At each iteration, the boost-clustering algorithm clusters data points that were hard to cluster in previous iterations. An important issue to be addressed here and that is the cluster correspondence problem between the clustering results of different iterations (Frossyniotis et al., 2004).

2.2 Feature Extraction

The first step in every clustering process is to extract the feature image bands. These features must contain useful information to discriminate between different regions of the surface. In our experiment we have used two types of features:

- The filtered first pulse range image using gradient
- Opening filtered last pulse range image

By our experiments, these two features have enough information to extract our objects of interest.

The normalized difference of the first and last pulse range images (NDDI) is usually used as the major feature band for discrimination of the vegetation pixels from the others. However, building boundaries also show a large value in this image feature. It is because when the laser beam hits the exposed surface it will have a footprint with a size in the range of 15-30 cm or more. So, if the laser beam hits the edge of a building, then part of the beam footprint will be reflected from the top roof of the building and the other part might reach the ground (Alharthy and Bethel, 2002). The high gradient response on building edges was utilized to filter out the NDDI image using equation 6.

$$NDDI = \frac{FPR - LPR}{FPR + LPR} \quad (6)$$

if gradient $\geq$ threshold, then (FPR-LPR) = 0.0

where
FPR = first-pulse range image data
LPR = last-pulse range image data

The gradient of an image is calculated using equation 7:

$$G(image) = \sqrt{G_x(image)^2 + G_y(image)^2} \quad (7)$$

where
G_x = gradient operators in x direction.
G_y = gradient operators in y direction.

The morphology Opening operator is utilized to filter elevation space. This operator with a flat structuring element eliminates the trend surface of the terrain. The main problem of using this filter is to define the proper size of the structuring element which should be big enough to cover all 3D objects which can be found on the terrain surface. The Opening operation is defined by:

$$A \circ B = (A \Theta B) \oplus B \quad (8)$$

where

$$A \oplus B = \left\{ x \mid \left(\hat{B}_x \cap A \right) \subseteq A \right\} \quad (9)$$

is the morphological Dilation of set A with structure element B. And

$$A \Theta B = \left\{ x \mid B_x \subseteq A \right\} \quad (10)$$

is the morphological Erosion of set A with structure element B (Gonzalez and Woods, 2006).

2.3 Quality Analysis

Comparative studies on clustering algorithms are difficult due to lack of universally agreed upon quantitative performance evaluation measures (Jain et al., 1999). Many similar works in the clustering area use the classification error as the final quality measurement; so in this research, we adopt a similar approach.

Here, we use error matrix as main evaluation method of interpretation result. Each column of this matrix indicates the instances in a predicted class. Each row represents the instances in an actual class. All the diagonal variants refer to the correct interpreted numbers of different classes found in reality. Some measures can be derived from the error matrix, such as producer accuracy, user accuracy and overall accuracy (Liu et al, 2007).

Producer Accuracy (PA) is the probability that a sampled unit in the image is in that particular class. User Accuracy (UA) is the probability that a certain reference class has also been labelled that class. Producer accuracy and user accuracy measures of each class indicate the interpretability of each feature class. We can see the producer accuracy and user accuracy of all the classes in the measures of "producer overall accuracy" and "user overall accuracy".

$$PA_i = \left(\frac{N_{i,i}}{N_{.i}} \right) * 100\% \quad , \quad UA_i = \left(\frac{N_{i,i}}{N_{i.}} \right) * 100\% \quad (11)$$

where
$N_{i,j}$ = (i,j)th entry in confusion matrix
$N_{i.}$ = the sum of all columns for row i
$N_{.j}$ is the sum of all rows for column i.

"Overall accuracy" considers all the producer accuracy and user accuracy of all the feature classes. Overall accuracy yields one number of the whole error matrix. It's the sum of correctly classified samples divided by the total sample number from user set and reference set (Liu et al, 2007).

$$OA = \frac{\sum_{i=1}^{k} N_{i,i}}{\left(\sum_{i=1}^{k} N_{.i} + \sum_{i=1}^{k} N_{i.} \right)} * 100\% \quad (12)$$

Another factor can be also extracted from confusion matrix to evaluate the quality of classification algorithms, which is K-

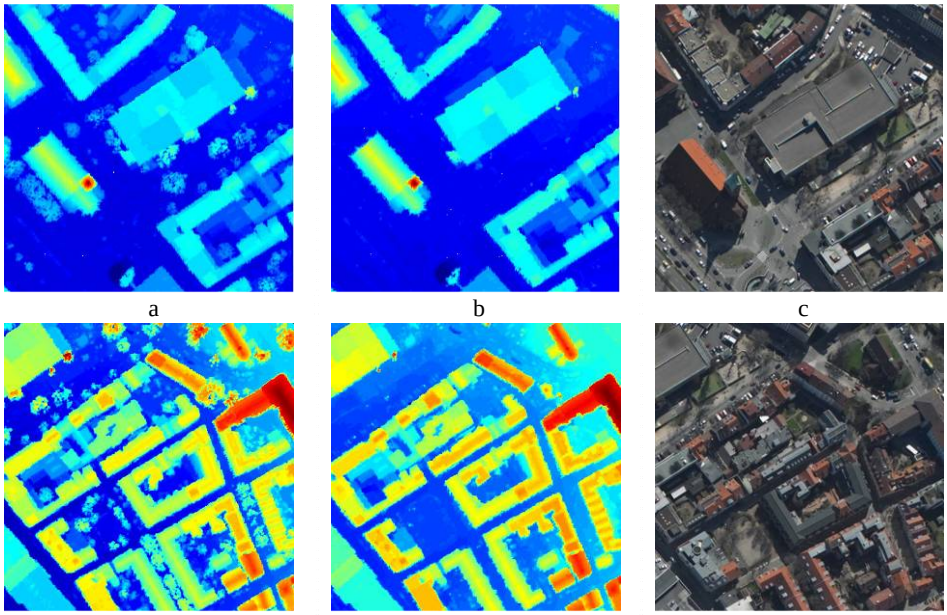
qualifier used to quantify the suitability of the whole clustering method.

$$K = \frac{\sum_{i=1}^{k}\sum_{i=1}^{k} N_{i,j} \cdot \sum_{i=1}^{k} N_{i,i} - \sum_{i=1}^{k} (N_{i.} \cdot N_{.i})}{\left(\sum_{i=1}^{k}\sum_{j=1}^{k} N_{i,j}\right)^2 - \sum_{i=1}^{k} (N_{i.} \cdot N_{.i})} \quad (13)$$

where
k = number of clusters

3. EXPERIMENTAL RESULTS

In this research, we have used two subsets of LiDAR data recorded from the city of Stuttgart, Germany. This data is recorded in four bands of first and last pulse range and intensity. The pixel size of this data is 30 cm. This means the average density of the recorded 3D points which is close to 9 per meter.



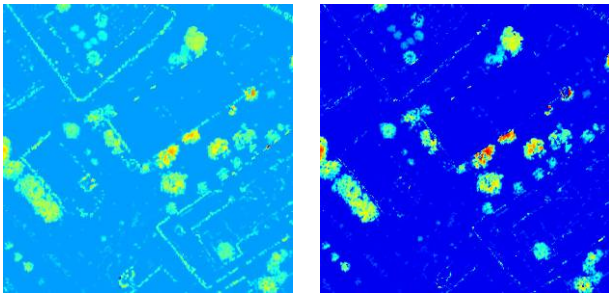
a b c d e f

Figure 1. Datasets used in our research. a) first pulse range of the first dataset, b) last pulse range of the first dataset, c) digital aerial image of the first dataset, d) first pulse range of the second dataset, e) last pulse range of the second dataset, f) digital aerial image of the second dataset

For better understanding of the objects, digital color (RGB) images have been also captured from this area using a medium format digital areal camera. In figure 1, color-coded first and last pulse images and also the RGB images of the investigated areas are illustrated. The trees can be distinguishes by comparing first and last pulse images.

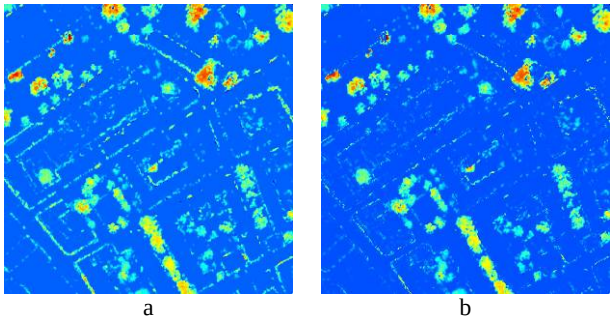
3.1 Results of Feature Extraction Algorithms

The level of the discrepancy between first and last return heights before and after applying the gradient filter is shown in figures 2, 3 for our two datasets. The discrepancy was larger than zero in the tree regions as expected.



a b

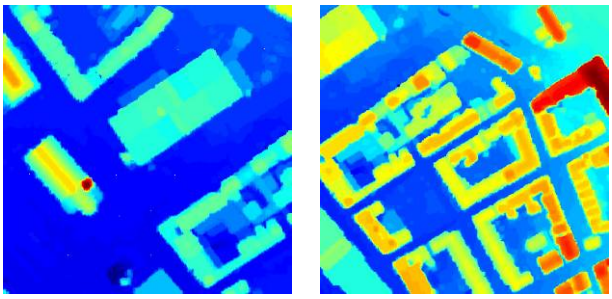
Figure 2. The normalized difference of the first and last pulse range images for our first dataset. a) before gradient filtering, b) after gradient filtering



a b

Figure 3. The normalized difference of the first and last pulse range images for our second dataset. a) before gradient filtering, b) after gradient filtering

The feature image of applying the morphological operator on last pulse range image with 5*5 structuring element is illustrated in figure 4. Here, the size of structuring element is selected by experiments on these two datasets.

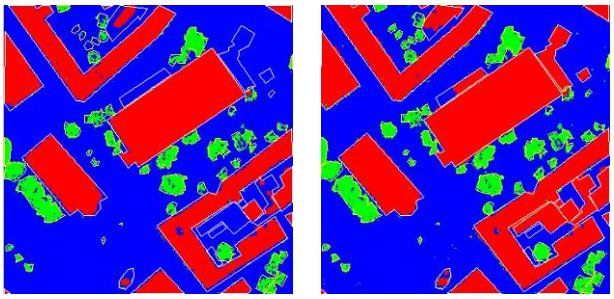


a b

Figure 4. Applying morphological opening operator with structuring element of size 5*5 to last pulse range images. a) the first dataset, b) the second dataset

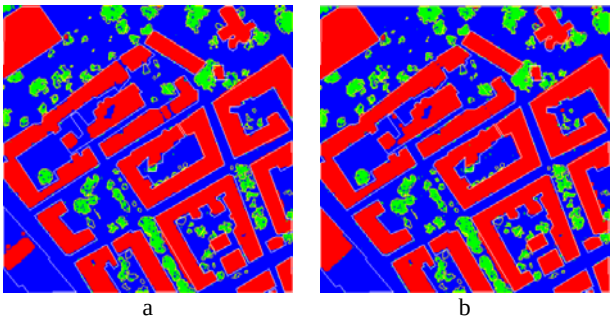
3.2 Evaluation of the Clustering Results

The results of k-means and boost k-means clustering algorithms applied to features of our two datasets are shown in figure 5 and figure 6. In our experiments the cluster number is considered fixed and equal to 3 because our objects of interest in urban areas are bare earth (blue), vegetation (green) and buildings (red). For the creation of confusion (error) matrix, first, the ground truth (also known as reference clustering results) should be defined. For this, 3D vectors of these areas consist of vegetation and building areas are used. The areas of polygons in pixel unit (number of pixels in the vector polygons of objects) are used as the values of reference clusters in error matrices. The user values are computed by counting the number of truly clustered patterns inside the polygons.



a b

Figure 5. Overlay of reference vectors on clustering results of first dataset. a) result of k-means algorithm, b) result of boost k-means algorithm.



a b

Figure 6. Overlay of reference vectors on clustering results of second dataset. a) result of k-means algorithm, b) result of boost k-means algorithm.

On the first view, both clustering algorithms provide reasonable classes of vegetation, buildings and ground but an accurate and numerical comparison will be carried out comparing the true object elements in the areas of interest.

In Tables 1, 2, the confusion matrices contain the number of pixels assigned to each cluster in the results of k-means clustering is provided. The confusion matrices and NMI factor of the results of boost k-means algorithm are also given in Tables 3, 4.

Table 1. Error matrix and quality factors of k-means clustering applied to first dataset.

Error Matrix		Reference Map		
		Building	Tree	Ground
Results	Building	34077	41	975
	Tree	205	6844	1178
	Ground	7946	2197	65607
Producer Accuracy		80.7%	75.4%	96.8%
Producer Accuracy		97.1%	83.2%	86.6%
Overal Accuracy		89.5%		
K-factor		0.801		

Table 2. Error matrix and quality factors of k-means clustering applied to second dataset.

Error Matrix		Reference Map		
		Building	Tree	Ground
Results	Building	58393	120	1858
	Tree	261	9808	1810
	Ground	10025	3809	68570
Producer Accuracy		85.0%	71.4%	94.9%
Producer Accuracy		96.7%	82.6%	83.2%
Overal Accuracy		88.4%		
K-factor		0.798		

It should be noted that the confusion matrix is should be diagonal in the ideal case. According to the above confusion matrices and NMI factors and also visual interpretation, improvement in results of clustering using boosting method is obvious for our classes of interest in theses datasets.

Table 3. Error matrix and quality factors of boost k-means clustering applied to first dataset.

Error Matrix		Reference Map		
		Building	Tree	Ground
Results	Building	39378	77	1895
	Tree	303	7757	1997
	Ground	2547	1248	63868
Producer Accuracy		93.2%	85.4%	94.2%
Producer Accuracy		95.2%	77.1%	94.4%
Overal Accuracy		93.2%		
K-factor		0.876		

Table 4. Error matrix and quality factors of boost k-means clustering applied to second dataset.

Error Matrix		Reference Map		
		Building	Tree	Ground
Results	Building	61027	212	1393
	Tree	428	10701	1178
	Ground	7224	2824	69667
Producer Accuracy		88.9%	77.9%	96.45
Producer Accuracy		97.4%	86.9%	87.4%
Overal Accuracy		91.4%		
K-factor		0.850		

4. SUMMARY

In this research a boost clustering methodology was applied on two datasets of LiDAR data in an urban area. The proposed method is a multiple clustering method based on the iterative application of a basic clustering algorithm. We evaluated this algorithm using two datasets, to investigate if this algorithm can lead to improved quality and robustness of performance. For the quality analysis of data clustering we used Some quality analysis factors such as produces, user and overall accuracy between the true labels and the labels returned by the clustering algorithms as the quality assessment measure. The experimental results on LiDAR datasets have shown that boost clustering algorithm can lead to better results compared to the solution obtained from the basic algorithm. The usefulness of the two feature channels Gradient Filtered NDDI and Opening of Last Pulse Range image for separating vegetation region with 3D extend and building regions from background has been also shown by the experiments.

There are also several directions for future work in this area. The most important is to determine the optimal number of clusters existing in the dataset. Other interesting future research topics concern the definition of best features of LiDAR data for data clustering and also using digital aerial and intensity images as well as the experimentation with other types of basic clustering algorithms and comparing the results of boost clustering with other strong clustering methods such as fuzzy k-means and neural networks or other multiple clustering based approaches.

REFERENCES

Alharthy,A., Bethel, J., 2002. Heuristic filtering and 3D feature extraction from LiDAR data. *IAPRS*, Graz, Austria. vol. XXXIII, pp. 29-35.

Bezdek, J., Pal, S., 1992. *Fuzzy Models for Pattern Recognition: Methods that Search for Structures in Data*. IEEE Press, New York, NY. 539 pages.

Dulyakarn, P., Rangsanseri, Y., 2001. Fuzzy c-means clustering using spatial information with application to remote sensing. In: *22nd Asian Conference on Remote Sensing, Singapore*. pp. 5-9.

Freund, Y., Schapire, Y., 1996. Experiments with a new boosting algorithm. In: *Proceedings of the Thirteenth International Conference on Machine Learning (ICML)*, pp: 148–156.

Freund, Y., Schapire, E., 1999. A short introduction to boosting. *Journal of Japanese Society for Artificial Intelligence*, 14(5), pp. 771-780.

Frossyniotis , D., Likas, A., Stafylopatis, A., 2004. A clustering method based on boosting. *Pattern Recognition Letters*, 25, pp. 641–654.

Gonzalez R. C., Woods, R. E. 2006. *Digital Image Processing* (3rd Edition), Prentice-Hall, Inc., Upper Saddle River, NJ, USA.

Jain, A.K., Dubes, R.C., 1988. *Algorithms for Clustering Data*, Prentice Hall, New Jersey.

Jain, A. K., Murty, M. N., Flynn, P.J., 1999. Data clustering: a review. *ACM Computing Surveys*, 31 (3). pp. 264-323

Kearns, M., Valiant. L. G., 1998. Learning boolean formulae or finite automata is as hard as factoring. Technical Report TR-14-88, Harvard University Aiken Computation Laboratory.

Kuncheva, L. I., 2004. *Combining Pattern Classifiers, Methods and Algorithms*. John Wiley & Sons, Inc., Hoboken, New Jersey. pp. 251-253.

Li, X., Wang, L., Sung, E., 2004. Improving adaBoost for classification on small training sample sets with active learning, In: *The Sixth Asian Conference on Computer Vision (ACCV)*, Korea.

Liu, C., Frazier, P., Kumar, L., 2007. Comparative assessment of the measures of thematic classification accuracy. *Remote Sensing of Environment*, 107, pp. 606-616.

Liu, Y., Jin, R., Jain, A. K., 2007. BoostCluster: boosting clustering by pairwise constraints. In: *KDD'07, the 13th International Conference on Knowledge Discovery and Data Mining*, San Jose, California, USA. pp: 450-459.

Pedrycz, W., 1997. Fuzzy clustering with partial supervision. In: *IEEE Transactions on Systems, Man, and Cybernetics*, Part B: Cybernetics, 27(5), pp. 787-795.

Saffari, A., Bischof, H., 2007. Clustering in a boosting framework. In: *Proceedings of Computer Vision Winter Workshop*, St. Lambrecht, Austria.

Zhong, S., Ghosh, D., 2003. A Unified framework for model-based clustering. *Journal of Machine Learning Research.* 4, pp. 1001-1037.

EXTRACTING BUILDING FOOTPRINTS FROM 3D POINT CLOUDS USING TERRESTRIAL LASER SCANNING AT STREET LEVEL

Karim Hammoudi, Fadi Dornaika and Nicolas Paparoditis

Université Paris-Est, Institut Géographique National, Laboratoire MATIS
73 Avenue de Paris, 94160 Saint-Mandé Cedex, France
{firstname.lastname}@ign.fr

KEY WORDS: 3D Point Cloud, Hough Transform, RANSAC Method, K-means Clustering, Laser Scanner, Building Footprint, Building Reconstruction, City Modeling.

ABSTRACT:

In this paper, we address the problem of generating building footprints using terrestrial laser scanning from a Mobile Mapping System (MMS). The MMS constitutes a fast and adapted tool to extract precise data for 3D city modeling. Urban environments evolve over time due to human activities and other factors. Buildings are constructed or destroyed and the urban areas are extended. Therefore, the structures of the cities are constantly modified. Currently, building footprints can be generated using aerial data. However, aerial based footprints lack precision due to the nature of the data and to the associated extraction methods. The use of MMS is proposed as an alternative to perform this complex task. In this work, we propose an operational approach for automatic extraction of accurate building footprints. We describe the challenges associated with the terrestrial laser raw data acquired in realistic and dense urban environments. After a filtering stage on the 3D laser cloud point, we extract and reconstruct the dominant facade planes by combining the Hough transform, the k-means clustering algorithm and the RANSAC method. The building footprint is then estimated from these dominant planes. Preliminary experimental results are presented and discussed. The assessments show that this approach is very promising for the automation of building footprints extraction.

1 INTRODUCTION

Nowadays, city modeling has become an important subject of research for architectural lasergrammetry, photogrammetry and computer vision communities. There is an increasing need for 3D building descriptions in urban areas in several fields of application like city planning and virtual tourism. Therefore many research activities on city modeling have focused on the automatic generation of 3D building models from aerial images. Most pipelines which have been developed recover the 3D shape of roof surfaces, but building ground footprints come from existing databases acquired by the digitization and vectorization of cadastral maps or from surveying measurements.

Initially, the building footprints are extracted either in an automatic way using the aerial data (Cheng et al., 2008), (Tarsha Kurdi et al., 2006) or in a manual way requiring many surveyors to make measurements in the terrain. However, these footprint databases sometimes do not exist (e.g., in less developed countries, etc.), can be very difficult to obtain (e.g., in areas with difficult access or prohibited overflights), or can even be of insufficient geometrical quality with respect to some applications. Moreover, the automatic building footprints extraction using aerial images is a hard task. Imprecise and/or incomplete focusing will affect the modeling process in the sense that the final 3D building model will lack accuracy and details.

Recent progress in technologies have allowed the development and the construction of devices for rapid acquisition of 3D cartographic terrestrial data with very high precision in urban environments. The Mobile Mapping System allows an easy coverage of large scale areas such as districts and cities. The feasibility of this kind of system has been demonstrated (Haala et al., 2008), and the usage of this device is increasingly widespread for applications like the conservation of patrimony (Baz et al., 2008) or visualization. Many works using terrestrial laser scanning are particularly focused on segmenting and texturing the building facades (Boulaassal et al., 2007), (Pu, 2008).

This ground-based modeling is thus unavoidable for some applications such as facade texturing where images acquired by a ground based system need to be registered relatively to the aerial 3D model to ensure a satisfactory mapping. Matching the street level images with the 3D aerial model is an extremely complex due to the generalization problems. The data acquired by ground-based 3D data collection systems, can be used to extract and model facades that can advantageously replace the ground footprints in the aerial reconstruction process, thus leading to a coherent use of both aerial and terrestrial data.

This paper focuses on the first step of a global 3D facade reconstruction framework, i.e. the extraction of the facade footprints and planes. The MMS constitutes an alternative and reliable tool which can be useful to obtain building footprints with very high accuracy and details. The aim of this study is to propose an operational approach for automated building footprints extraction in urban environments. The remainder of the paper is organized as follows: Section 2 states the problems related to the raw laser data and their processing. Section 3 presents the proposed approach for extracting the building's footprints and facade planes. Section 4 gives some promising experimental results.

2 OVERVIEW ON PROBLEMS RELATED TO THE LASER RAW DATA AND THEIR PROCESSING

In this study, we use a mobile mapping system for acquiring georeferenced 3D laser point clouds. The Terrestrial Laser Scanning system (TLS system) is a 2D profile scanner. The third dimension is induced by the vehicle displacement. In addition to this, the Mobile Mapping System is equipped with a Global Positioning System (GPS), an Inertial Measurement Unit (IMU) and a Distance Measuring Instrument (DMI), namely an odometer. This

equipment was precisely installed by topometry. It allows the gathering of georeferenced 3D laser data with a very high density and also much information about the acquisition (see section 4).

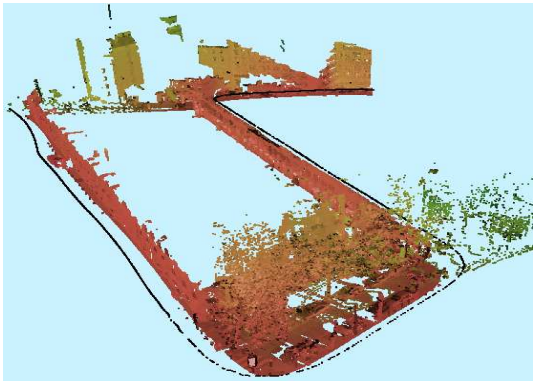


Figure 1: Visualization of the 3D point cloud of very high density. This raw data represents the building facade acquisition. The black line represents the trajectory of the laser sensor.

The laser data are acquired under **realistic conditions** in dense urban environments. Moreover, the datasets are collected in a particular container related to the laser scanner, particularly in a range of 3D points. In this context, the data must be manipulated with much precaution. Here we will describe the main difficulties associated with 3D data.

- **The acquisition:** The laser sensor data can be represented in two ways; either as a container with data organized linearly in temporal sequences of 201 points (data frames) or like a cloud of 3D points (georeferenced data). The conditions of acquisition are very variable in realistic and dense urban environments. Mobile objects cause a thickening of the acquired cloud when the vehicle stops. Certain acquired points model an ephemeral surface and could be considered as erroneous points. Moreover, the density of the facades vary according to the speed of the vehicle.
- **The occlusions:** pose a problem for the complete acquisition of building facades in urban environments. The occlusions could be caused by two categories of obstacles, static or dynamic created by man-made and natural objects. The raw cloud may suffer from missing data due to the presence of pedestrians, trees, mobile and parked vehicles and many others objects (see figure 2). Alas, this inevitable phenomenon affects the modeling process.
- **The laser reflectance:** could cause confusions in the 3D data interpretation. Certain points don't model a physical surface. This effect appears on a retroreflector surface. Observations sometimes show an aureole of points around road signs. These dispatched points represent erroneous data. Moreover, certain points model a different surface other than the surface of interest. Sometimes, the beam of the laser either rebounds off of the outside of the window or it passes through the window and models the inside of the dwelling. These scattered points represent erroneous information for the facades modeling. In addition to this, other less frequent effects could arise due to poorly reflective surfaces.
- **The redundancy of data:** is due to many factors. The acquisition is continuous even when the vehicle is stopped. We have adopted this strategy to facilitate the acquisition of a large area and to use data as common bases for our different projects. Therefore, raw laser data may contain many redundant frames. Moreover, due to sensor characteristics (orientation and linear scanning), we could sometimes have up to three acquisitions of the same facade part caused by the graining of the laser beam in the turns. The redundancy of data (points, frames, parts of the facade) presents an inconvenience for the feature extraction techniques based on vote schemes or random trials.

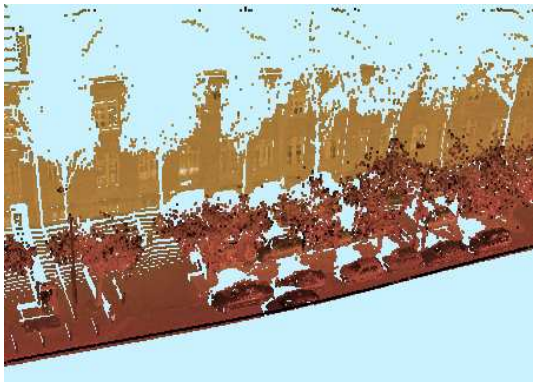


Figure 2: A street in the city of Paris. The building facade is partially occluded by trees and parked vehicles.

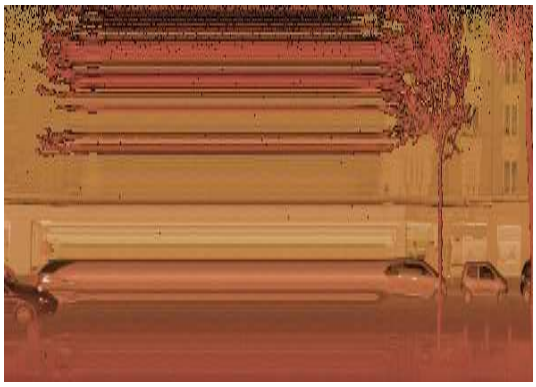


Figure 3: Returned intensities of the 2D scans. The redundancy effect appears when the vehicle temporarily stops. The vehicle and the branches seem stretched.

This brief description allows us to acknowledge several problems associated with the raw laser data. The 3D data should undergo several preprocessing steps before becoming exploitable. Thus we need a process robust to some outliers and noisy data.

3 PROPOSED APPROACH

In this section, we describe our approach which consists of two stages. The first stage focuses on the 3D cloud points preprocessing. The second stage aims at the building footprint extraction. In this work, we assume that buildings have simple polygonal shapes.

3.1 3D data preprocessing

3.1.1 Partial filtering of redundant points As we have mentioned earlier, the laser sensor constantly sweeps the building facade even when the vehicle is stopped. Consequently, the acquired raw data may contain many redundant frames due to this continuous acquisition. For this reason, we have defined a measure between two consecutive frames based on point-to-point distances. The redundant frames are thus detected and removed from

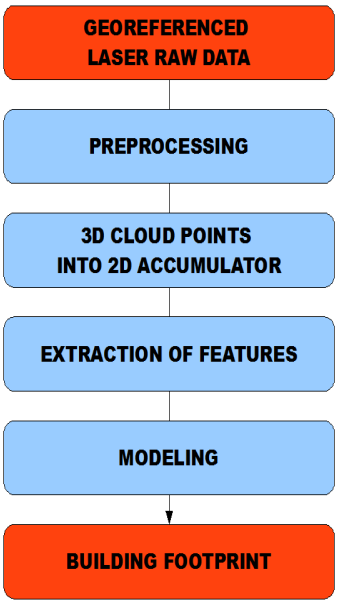


Figure 4: Data flow diagram of the building footprint extraction approach.

the dataset by thresholding these distances. Moreover, this temporal effect distinctly appear on the image of intensity of the laser beam and looks like a succession of rows having the same intensity (see figure 3). Therefore detecting redundant frames can also be based on differences between the intensity of return of two consecutive rows. This step solves only the problem of data redundancy related to the sensor immobility.

3.1.2 Volume of interest The sensor characteristics are used to select the 3D points belonging to the facade plane. The 3D available points are principally positioned above the vehicle to reduce the problem of occlusions. A horizontal band is defined between two horizontal planes. The lower plane passes through the sensor center. The upper plane is shifted by a certain distance that is related to the height of the buildings under study. We precise that the ground altitude could be simply deduced by measuring manually the laser sensor height. Finally, the volume of interest is defined by the georeferenced trajectory of the vehicle and the above horizontal band. The 3D points not included in this volume will be removed from the dataset.

3.1.3 Exploiting the linearity of 3D data After the preceding filtering steps, the frames have undergone a horizontal cropping. The data structure represented by frames is now represented by a sequence of 3D points. We exploit the fact that in this representation facade points are locally aligned. We seek facade points which are principally organized vertically. Thus, the dataset in this sequence is parsed by triplets. The central point of each triplet is kept in the dataset if the triplet is aligned, otherwise it will be removed from the dataset. Therefore, the coplanar points of the building facade are kept. Besides, we observe that the 3D points belonging to other linear structures are also kept.

3.1.4 Mapping the 3D point cloud onto a 2D accumulator The goal of this step is twofold. Firstly, it aims at removing noisy and outlier points. Secondly, it gives a very compact representation of the filtered 3D points. Since we are interested in the vertical structures that generally represent the facades, we project the 3D cloud on a horizontal plane. More precisely, the 3D points are projected into a 2D grid to create an accumulation space. Each point of the cloud votes in one cell, giving a score. Only cells having a high score are kept. The process uses a global threshold which is compared to the maximum score. By this technique, the erratic points of the cloud are removed from the data. The cells with high scores are principally facade points with high density.

Several techniques for the detection of outliers in laser point clouds can be found in (Sotoodeh, 2006) and (Sotoodeh, 2007).

3.2 Building footprints extraction

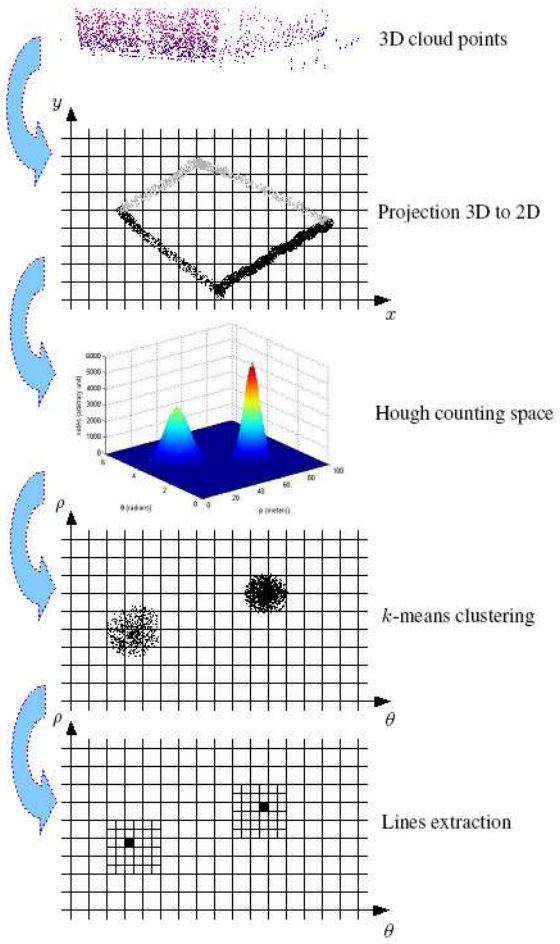


Figure 5: The main steps of our proposed approach for building footprint extraction.

The goal is to automatically extract the building footprint using the 3D filtered points cloud contained in the compact 2D accumulator. The building footprint is a set of 2D segments that can be detected in this 2D space. Recall that the vertical structure of the facades is captured by the scores of the cells. Each cell contains, if any, a set of 3D points $P(x,y,z)$. Furthermore, an efficient extraction can be obtained by working with the barycenters (2D coordinates) of the cells together with their scores. Our approach combines the use of the counting space of Hough Transform, the k-means clustering technique and the RANSAC method. We briefly describe these three techniques and their properties applied in our context. Figure 5 illustrates the main extraction steps.

The Standard Hough Transform (SHT) allows the extraction of the 2D lines among 2D dataset points (Hough, 1962). In the field of our application, this method is currently used to detect the building boundaries in aerial images using the edge points. This method is also used to extract buildings in LIDAR data (e.g., (Tarsha Kurdi et al., 2007) and (Karsli and Kahya, 2008)).

In our approach, we only use the voting steps associated with the Hough transform. We describe briefly here the principle. We made a Hough accumulation space in the discretized parameter space ρ and θ. Each 3D point $P(x,y,z)$ of the dataset (facade points) votes in all cells of the Hough space accumulation verifying the following constraint:

$$\rho = x \cdot \cos\theta + y \cdot \sin\theta \tag{1}$$

where ρ is a length of the normal of the line to the origin and θ is the orientation associated to the normal vector. Each couple (ρ,θ) is unique if $\theta \in [0,2\pi]$ and $\rho \geq 0$.

The cells containing a high score correspond more or less to a facade. However, we aim at determining precisely and automatically the best fit lines of points characterizing the building facades (see figure 6). In our case, we do not carry out the lines extraction step that is based on the local maxima values of the Hough accumulation space since this requires a very difficult tuning of some parameters.

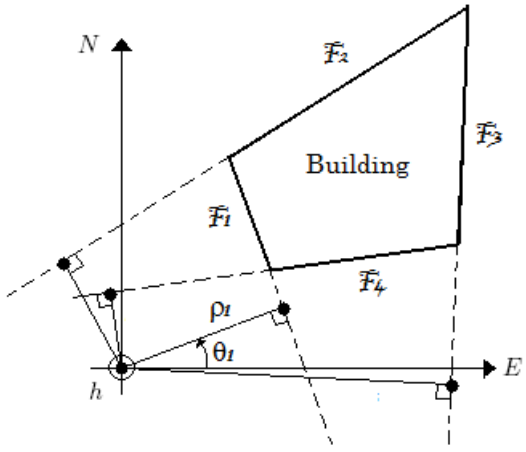


Figure 6: The usage of the Hough transform for extracting building facade lines when the different facades have similar densities.

The local maxima often provide approximate or erroneous solutions. Any partially occluded facade has a low density of 3D points and has thus a low vote in the Hough counting space compared to the non-occluded facade. For this reason, if the vote threshold is too low, many lines will be extracted. Inversely, if the threshold is too high, many lines could be missed. In addition to this, the line estimation depends also on the discretization steps of ρ and θ values. Close lines characterizing the approximation of the same potential line are sometimes extracted. The usage of a large neighborhood to determine the local maxima in the Hough space could reduce this effect. Nevertheless, tuning the threshold is a very difficult task in many cases.

We remind that our approach deals with buildings having simple polygonal footprints. The discretization steps depend also of the building characteristics. The ρ step is related to the minimal distance defined between two facades planes with a similar orientation, the θ step is related to the minimal angle defined between two adjacent facades.

In the urban context, certain characteristics of the building facades are a priori known. The number of facades of the buildings is known in advance (between 3 and 10). Our idea is to use a k-means clustering algorithm to replace the detection of local maxima in the accumulation space—the parameter space of θ and ρ values, in order to automatically determine the exact number of facades and their support 2D lines.

The k-means algorithm is a well-known unsupervised clustering method commonly used to cluster n objects of the input dataset into k homogeneous partitions, $k<n$, for example (Forgy, 1965) and (Macqueen, 1967). We use this technique in a classic way. Nevertheless, several other various clustering techniques exist and a survey is found in (Xu and Wunsch, 2005). Mathematically, the k clusters are determined by minimizing an objective function such as the sum of the squared distances between the points and the corresponding centroids such as:

$$intra\_distance = \sum_{i=1}^{k} \sum_{P_m \in S_i} \left((\rho,\theta)_{(P_m)} - (\bar{\rho}_i, \bar{\theta}_i)\right)^2 \tag{2}$$

where $(\rho,\theta)_{(P_m)}$ is the value of (ρ,θ) associated with all 3D points $P_m(x_m, y_m, z_m)$ included in the corresponding cell, k is the number of clusters S_i, $i = 1, 2, \ldots, k$, and $(\bar{\rho}_i, \bar{\theta}_i)$ is the centroid of the cluster S_i. The above score is simply the intra-cluster distance measure. More specifically, the score is calculated only for the cells containing a strictly positive vote in the Hough space. Besides, the sum in the above equation is carried out for all 3D point candidates even if they vote all in the same cell.

We can also measure the inter-cluster distance, or the distance between clusters, which we want to be as big as possible. This measure is given by

$$inter\_distance = \min\left((\bar{\rho}_j, \bar{\theta}_j) - (\bar{\rho}_i, \bar{\theta}_i)\right)^2, \quad i \neq j \tag{3}$$

Since we want both of these measures to help us determine if we have a good clustering, i.e., a clustering which results in compact clusters which are well separated, we must combine them in some way. The obvious way is to minimize the following objective function:

$$validity = \frac{intra\_distance}{inter\_distance} \tag{4}$$

In our case, the k-means algorithm is run for each k value belonging to the predefined interval. Each run provides a score based on (4). The potential number of facades is the k value corresponding to the minimum of these scores. This validity measure for the determination of the number of clusters in k-means clustering was proposed in (Ray and Turi, 1999). Thus the number of facades could be known even when the facades have heterogeneous densities of 3D points.

More precisely, when one run is carried out for a given k, the algorithm is not guaranteed to return the global optimum because the convergence depends on the initial seeds selected. The k-means algorithm is extremely fast. For this reason, a method which is commonly employed is to run the algorithm several times and select the best clustering available for each k-value. In our case, the first run is carried out by setting the initial seeds to the local maxima in the Hough counting space. The other runs are randomly initialized inside the Hough counting space.

Now that the number of clusters k is known, one can compute a 2D line solution (facade support) for each detected cluster of points. Several solutions can be used to model the facade such as the use of the centroid of each cluster, or the solution with the

highest vote for each cluster. We propose to employ a more accurate method such as the RANSAC method to model the building footprint. One advantage of our approach is the following. A curved facade will be approximated by a single line. In addition to this, much information deduced by the clustering step allow to automatically adjust the parameters of the RANSAC algorithm and to thus improve the precision of the detected lines.

The RANSAC method is commonly used to detect lines among edge points (Bretar and Roux, 2005) and (Sester and Neidhart, 2008). We use a classic method (Fischler and Bolles, 1981). For each detected cluster, we use the following process. We use the original space of parameter (x,y). Two different points belonging to the cluster are randomly selected, characterizing a line. A neighborhood is defined along this line by a minimal distance between a point and the line. This process is iteratively repeated until the number of inlier points in the neighborhood is maximized. In our approach, the number of facade points is roughly equal to the number of points for each cluster. Furthermore, the minimal distance associated with the RANSAC technique can be determined from the dispersion of the cluster. When this step is carried out, the best fit lines of 2D points are extracted using the Least Squares Adjustment (LSA technique). A set of 2D segments giving the building footprint is then obtained from the detected 2D lines.

4 PRELIMINARY RESULTS

The acquired 3D data correspond to the facades of buildings in the 12th district of the city of Paris. In this study, we use a high precision 2D laser sensor LMS-Q120i made by RIEGL company [1]. The laser sensor is positioned on the roof of the vehicle. Its beam plane is perpendicular to the vehicle trajectory. The system allows us to carry out 10000 measurements per second and the beam vertically sweeps with an opening of 80 degrees (-20 to 60 degrees with respect to the horizontal). The angular precision of the beam is equal to 0.01 degrees. More specifically, the precision of laser-based measurements is approximately 3 cm at 150 m. In this study, the angular resolution was configured to 201 points by frame (see figure 7). The ground based laser range transmits laser pulses with simple echo.

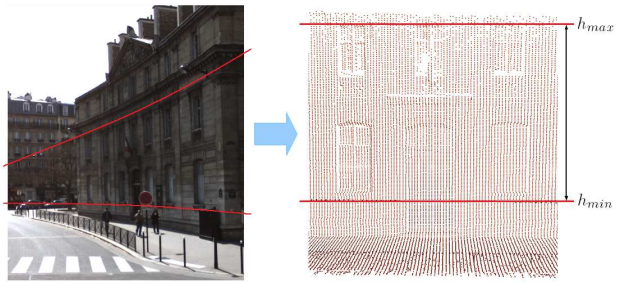


Figure 7: Acquisition of the 3D point cloud using the 2D laser sensor. The frame shows a selected band without occlusions.

The raw measurements provided by the laser sensor are points that are parameterized by distance and angle. The reflected intensity of the laser is between 0 and 1. The coordinates of the 3D points are expressed in the laser sensor coordinate system and also in a common coordinate system, namely the ground reference (absolute) Northern, Eastern and Altitude in Lambert 93. The precision of a 3D point is not easy to evaluate because it depends on the laser precision and on the referencing system precision.

[1] Link to RIEGL company: *http://www.riegl.com*

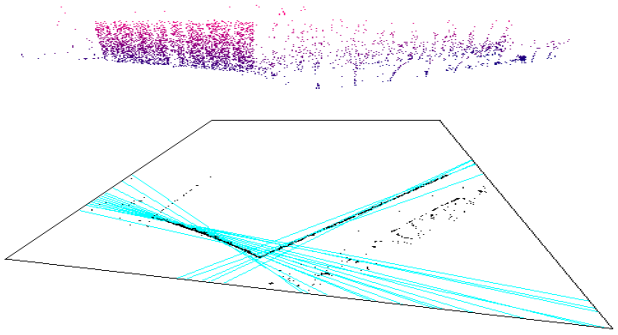


Figure 8: Two difficult facades for the classical Hough Transform: i) a curved facade, and ii) a facade with a low density of 3D points. The detected lines correspond to the local maxima of the Hough space accumulation using the filtered cloud.

The experiments are carried out on two building facades having different architecture and different density of acquired 3D points. One can thus assess the robustness and the efficiency of the proposed approach.

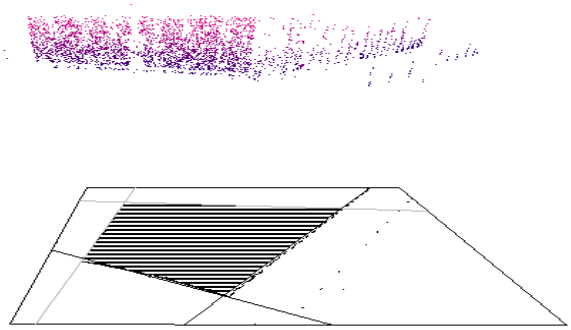


Figure 9: Extracting the building footprint lines using the proposed approach.

Figures 8 and 9 show the extraction of the building footprint lines using the classical Hough Transform and our proposed approach, respectively. The 3D point cloud is presented in the upper part of the figure. The projection onto the 2D accumulator is presented in the lower part of the figure. The studied building illustrates two difficult cases for a classical Hough Transform. Indeed, the left facade does not suffer from occlusions but it is slightly curved. On the other hand, the right facade which is a planar structure is partially occluded, that is, the density of its 3D points in the 2D accumulator space is much lower than that of the left facade.

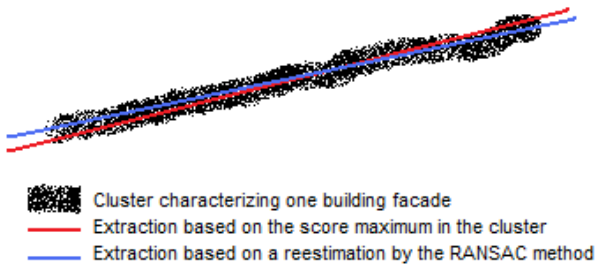
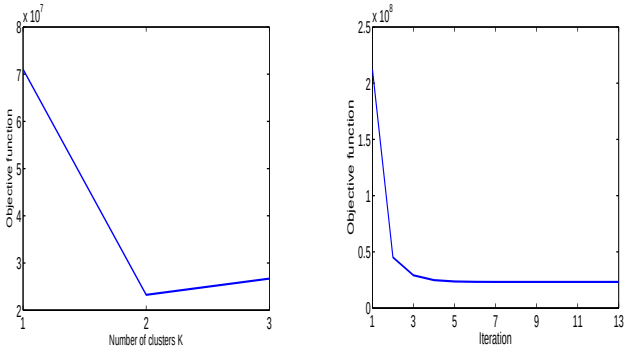


Figure 10: Comparative schema illustrating the precision of lines detection step on one simulated facade.



(a) **(b)**

Figure 11: The application of the k-means clustering.

As can be seen, the classical Hough Transform provided many 2D lines (facade support) corresponding to the many local maxima in the Hough counting space. We can observe that both the curved facade and the partially occluded facade are modelled by several lines. However, by using our proposed approach based on k-means clustering, the correct and accurate 2D lines were obtained. As explained above, the 2D lines can be given either by the centroid of the cluster, its maximum or by the RANSAC technique. As can be seen in figure 10, the building footprint extraction will be more precise using the RANSAC method. The maximum score method detects the line comprising the maximum of points, but it is not necessarily the correct 2D line. The information provided by the clustering method allows us to refine the estimation of the facade lines by exploiting the number of points and the dispersion if the detected cluster (facade) within the RANSAC framework.

Figure 11 shows the application of the k-means clustering algorithm on the 3D data associated with the two facades. Figure 11.**(a)** depicts the validity score as a function of the number of clusters k. As can be seen the optimal value for k is 2. Figure 11.**(b)** shows the convergence associated with this optimum. The footprint lines extracted from this clustering are illustrated in figure 9.

5 CONCLUSIONS AND FUTURE WORK

We presented an approach for the automatic extraction of the building footprint in urban environments. This approach does not require previous knowledge of the number of facades in the input dataset. Moreover, the approach is robust to the heterogeneous densities of facade points. The proposed approach is based on fast filtering and feature extraction techniques. This stage constitutes an essential task for 3D building modeling. Experimental results show the feasibility and robustness of the proposed approach on small islets of buildings.

Future work may investigate the extension of the approach to buildings with a high complexity of shapes and the possibility of application to large areas because each islet of the buildings is delimited by its georeferenced trajectory. Furthermore, since outdoor squares inside the buildings are inaccessible areas for the vehicle, we plan to extend our approach to model full buildings by exploiting the terrestrial data and the corresponding aerial data.

ACKNOWLEDGEMENT

The authors would like to thank Bertrand Cannelle from IGN for his assistance with software and helpful discussions related to the data used in this work.

REFERENCES

Baz, I., Buyuksalih, G., Kersten, T. and Jacobsen, K., 2008. Documentation of istanbul historic peninsula by static and mobile terrestrial laser scanning. In: ISPRS08, p. B5: 993 ff.

Boulaassal, H., Landes, T., Grussenmeyer, P. and Tarsha Kurdi, F., 2007. Automatic segmentation of building facades using terrestrial laser data. In: Laser07, p. 65.

Bretar, F. and Roux, M., 2005. Extraction of 3D planar primitives from raw airborne laser data: a normal driven ransac approach. In: MVA, pp. 452–455.

Cheng, L., Gong, J., Chen, X. and Han, P., 2008. Building boundary extraction from high resolution imagery and lidar data. In: ISPRS08, p. B3b: 693 ff.

Fischler, M. A. and Bolles, R. C., 1981. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Commun. ACM 24(6), pp. 381–395.

Forgy, E. W., 1965. Cluster analysis of multivariate data: efficiency vs interpretability of classifications. Biometrics 21, pp. 768–769.

Haala, N., Peter, M., Kremer, J. and Hunter, G., 2008. Mobile lidar mapping for 3D point cloud collection in urban areas: A performance test. In: ISPRS08, p. B5: 1119 ff.

Hough, P., 1962. Method and means for recognizing complex patterns. In: US Patent.

Karsli, F. and Kahya, O., 2008. Building extraction from laser scanning data. In: ISPRS08, p. B3b: 289 ff.

Macqueen, J., 1967. Some methods for classification and analysis of multivariate observations. In: 5th Berkeley Symposium on Mathematical Statistics and Probability, Berkeley, CA.

Pu, S., 2008. Generating building outlines from terrestrial laser scanning. In: ISPRS08, p. B5: 451 ff.

Ray, S. and Turi, R., 1999. Determination of number of clusters in k-means clustering and application in colour image segmentation. In: Proceedings of the 4th International Conference on Advances in Pattern Recognition and Digital Techniques, pp. 137–143.

Sester, M. and Neidhart, H., 2008. Reconstruction of building ground plans from laser scanner data. In: AGILE08.

Sotoodeh, S., 2006. Outlier detection in laser scanner point clouds. In: International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences XXXVI-5, pp. 297–302.

Sotoodeh, S., 2007. Hierarchical clustered outlier detection in laser scanner point clouds. In: Laser07, p. 383.

Tarsha Kurdi, F., Landes, T. and Grussenmeyer, P., 2007. Hough-transform and extended ransac algorithms for automatic detection of 3D building roof planes from lidar data. In: Laser07, p. 407.

Tarsha Kurdi, F., Landes, T., Grussenmeyer, P. and Smigiel, E., 2006. New approach for automatic detection of buildings in airborne laser scanner data using first echo only. In: PCV06.

Xu, R. and Wunsch, D., 2005. Survey of clustering algorithms. Neural Networks, IEEE Transactions on 16(3), pp. 645–678.

DETECTION OF BUILDINGS AT AIRPORT SITES USING IMAGES & LIDAR DATA AND A COMBINATION OF VARIOUS METHODS

Demir, N.[1], Poli, D.[2], Baltsavias, E.[1]
1- (nusret,manos@geod.baug.ethz.ch)
Institute of Geodesy and Photogrammetry, ETH Zurich, CH-8093, Zurich, Switzerland
2- (daniela.poli@jrc.ec.europa.eu)
European Commission - Joint Research Center, Ispra (VA), Italy

KEY WORDS: DTMs/DSMs, Lidar Data Processing, Multispectral Classification, Image Matching, Information Fusion, Object Detection, Buildings

ABSTRACT:

In this work, we focus on the detection of buildings, by combining information from aerial images and Lidar data. We applied four different methods on a dataset located at Zurich Airport, Switzerland. The first method is based on DSM/DTM comparison in combination with NDVI analysis (Method 1). The second one is a supervised multispectral classification refined with a normalized DSM (Method 2). The third approach uses voids in Lidar DTM and NDVI classification (Method 3), while the last method is based on the analysis of the density of the raw Lidar DTM and DSM data (Method 4). An improvement has been achieved by fusing the results of the different methods, taking into account their advantages and disadvantages. Edge information from images has also been used for quality improvement of the detected buildings. The accuracy of the building detection was evaluated by comparing the results with reference data, resulting in 94% detection and 7% omission errors for the building area.

1. INTRODUCTION

In this work, we focus on the building detection for airport sites. The acquisition of a reliable geospatial reference database of airports, and in particular the automatic extraction of buildings and obstacles at airports, both have a critical role for aviation safety. Often, 3D information of airports is not available, not accurate enough, not complete, or not updated. Thus, methods are needed for generation of accurate and complete 3D geodata with high degree of automation. In particular, buildings and trees are considered as obstacles, so they should be correctly extracted. In this work, we focus on the detection of buildings, as a first step for their 3D extraction. There are several methods applied for this purpose, based on image and/or airborne Lidar data. In our approach, buildings are detected in aerial images and Lidar data through multiple methods using multispectral image classification, DSM (Digital Surface Model) and DTM (Digital Terrain Model) comparisons and density analysis of the raw Lidar point cloud. The detection quality is improved by a combination of the results of the individual methods. This paper will give a brief overview of the related work on this subject. Then, after the description of the test area at Zurich Airport, Switzerland, the strategy and methodology will be presented and the results will be reported, compared and commented. This work is a part of the EU 6th Framework project PEGASE (Pegase, 2009).

2. PREVIOUS WORK

Aerial images and Lidar data are common sources for object extraction. In digital photogrammetry, features of objects are extracted using 3D information from image matching or DSM/DTM data, spectral, textural and other information sources. Pixel-based classification methods, either supervised or unsupervised, are mostly used for land-cover and man-made structure detections. For the classical methods e.g. minimum-distance, parallelepiped and maximum likelihood, detailed information can be found in (Lillesand and Kiefer, 1994).

In general, the major difficulty in using aerial images is the complexity and variability of objects and their form, especially in suburban and densely populated urban regions (Weidner and Foerstner, 1995).

Regarding Lidar, building and tree extraction is basically a filtering problem in the DSM (raw or interpolated) data. Some algorithms use raw data (Sohn and Dowman, 2002; Roggero, 2001; Axelsson, 2001; Vosselman and Maas, 2001; Sithole, 2001; Pfeifer et al., 1998), while others use interpolated data (Elmqvist et al., 2001; Brovelli et al., 2002; Wack and Wimmer, 2002). The use of raw or interpolated data can influence the performance of the filtering. The algorithms differ also in the number of points they use at a time. In addition, every filter makes an assumption about the structure of bare-earth points in a local neighbourhood. This assumption forms the concept of the filter (Sithole and Vosselman, 2003). The region-based methods use mostly segmentation techniques, like in Brovelli et al. (2002), or using Hough transformation (Tarsha-Kurdi et al., 2007). Some researchers use 2D maps as prior information for building extraction (Brenner, 2000; Haala and Brenner., 1999; Durupt and Taillandier, 2006; Schwalbe et al., 2005). Topographic maps provide outlines, classified polygons and topologic and 2D semantic information (Elberink and Vosselman, 2006).

In general, in order to overcome the limitations of image-based and Lidar-based techniques, it is of advantage to use a combination of these techniques. Sohn and Dowman (2007) used IKONOS images to find building regions before extracting them from Lidar data. Straub (2004) combines information from infrared imagery and Lidar data to extract trees. Rottensteiner et al. (2005) evaluate a method for building detection by the Dempster-Shafer fusion of Lidar data and multispectral images. They improved the overall correctness of the results by fusing Lidar data with multispectral images.

Few commercial software packages allow automatic terrain, tree and building extraction from Lidar data. In TerraSCAN, a TIN is generated and progressively densified, the extraction of off-terrain points is performed using the angles between points to make the TIN facets and the other parameter is the distance to nearby facet nodes (Axelsson, 2001). In SCOP++, robust methods operate on the original data points and allow the simultaneous elimination of off-terrain points and terrain surface modelling (Kraus and Pfeifer, 1998).

In summary, most approaches try to find objects using single methods. In our strategy, this study suggests complying different methods using all available data with the focus on improving the results of one method by exploiting the results from the remaining ones.

3. INPUT DATA AND PREPROCESSING

The methods presented in this paper have been tested on a dataset of the Zurich airport. The available data for this region are: 3D vector data of airport objects, colour and CIR (Colour InfraRed) images, Lidar DSM/DTM data (raw and grid interpolated). The characteristics of the input data can be seen in Table 1.

Image Data	**RGB**	**CIR**
Provider	Swissphoto	Swissphoto
Scale	1: 10‘000	1: 6‘000
Scan Resolution	14.5 microns	14.5 microns
Acquisition Date	July 2002	July 2002
Ground Sampling Distance (GSD) (cm)	14.5 cm	8.7 cm
Lidar Data	**DSM**	**DTM**
Provider	Swisstopo	Swisstopo
Type	Raw & grid	Raw & grid
Raw point density & Grid Spacing	1 pt / 2 sqm & 2m	1 pt / 2 sqm & 2m
Acquisition Date	Feb. 2002	Feb. 2002
Vector data	Only for validation purposes	
Provider	Unique Co.	
Horizontal / Vertical Accuracy (2 sigma)	20 / 25 cm	

Table 1. Input data characteristics.

The 3D vector data describe buildings (including airport parking buildings and airport trestlework structures). It has been produced from stereo aerial images using the semi-automatic approach with the CC-Modeler software (Gruen and Wang, 1998). Some additional reference buildings outside the airport perimeter were collected using CIR images with stereo measurement by using LPS software. The images have been firstly radiometrically preprocessed (noise reduction and contrast enhancement), then the DSM was generated with the software package SAT-PP, developed at the Institute of Geodesy and Photogrammetry, ETH Zurich (Zhang, 2005). For the selection of the optimum band for matching, we considered the GSD, and the quality of each spectral channel based on visual checking and histogram statistics. Finally, the NIR band was selected for DSM generation. The final DSM was generated with 50cm grid spacing. Using this DSM, CIR orthoimages were produced with 12.5cm ground sampling distance. Lidar raw data (DTM and DSM) have been acquired with “leaves off”. The DSM point cloud includes all Lidar points (including points on terrain, tree branches etc.). The DTM data includes only points on the ground, so it has holes at building positions and less density at tree positions. The height accuracy (one standard deviation) is 0.5 m generally, and 1.5 m at trees and buildings, the latter referring only to the DSM. The grid DSM and DTM were interpolated from the original raw data by Swisstopo with the Terrascan commercial software.

4. BUILDING DETECTION

Four different approaches have been applied to exploit the information contained in the image and Lidar data, extract different objects and finally buildings. The first method is based on DSM/DTM comparison in combination with NDVI (Normalised Difference Vegetation Index) analysis for building detection. The second approach is a supervised multispectral classification refined with height information from Lidar data and image-based DSM. The third method uses voids in Lidar DTM and NDVI classification. The last method is based on the analysis of the density of the raw DSM Lidar data. The accuracy of the building detection process was evaluated by comparing the results with the reference data and computing the percentage of data correctly extracted and the percentage of reference data not extracted.

4.1 DSM/DTM and NDVI (Method 1)

By subtracting the DTM from the DSM, a so-called normalized DSM (nDSM) is generated, which describes the above-ground objects, including buildings and trees. As DSM, the surface model generated by SAT-PP and as DTM the Lidar DTM grid were used. NDVI image has been generated using the NIR and R bands. A standard unsupervised (ISODATA) classification was used to extract vegetation from NDVI image. The intersection of the nDSM with NDVI should correspond to trees. By subtracting the resulting trees from the nDSM, the buildings are obtained. 83% of building class pixels were correctly classified, while all of 109 buildings have been detected but not fully, the omission error is 7% . Within the detected buildings, some other objects, such as aircrafts and vehicles, were included. The extracted buildings are shown in Figure 1.

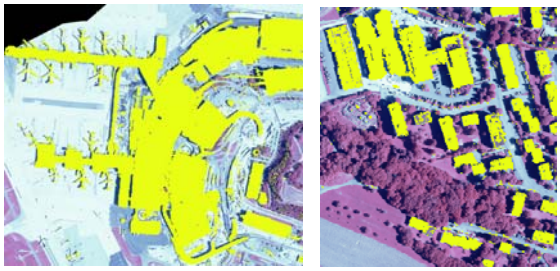


Figure 1. Building detection result from method 1. (Left: airport buildings, Right: residential area).

4.2 Supervised classification and use of nDSM (Method 2)

The basic idea of this method is to combine the results from a supervised classification with the height information contained in the nDSM. Supervised classification methods are preferable to unsupervised ones, because the target of the project is to detect well-defined standard target classes (airport buildings, airport corridors, bare ground, grass, trees, roads, residential houses, shadows etc.), present at airport sites. The training areas were selected manually using AOI (Area Of Interest) tools within the ERDAS Imagine commercial software (Kloer, 1994). Among the available image bands for classification (R, G and B from colour images and NIR, R and G bands from CIR images), only the bands from CIR images were used due to their better resolution and the presence of NIR channel (indispensable for

vegetation detection). In addition, new synthetic bands were generated from the selected channels: a) 3 images from principal component analysis (PC1, PC2, PC3); b) one image from NDVI computation using the NIR-R channels and c) one saturation image (S) obtained by converting the NIR-R-G channels in the IHS (Intensity, Hue, Saturation) colour space. The separability of the target classes was analyzed through use of plots by mean and standard deviation for each class and channel and divergence matrix analysis of all possible combinations of the three CIR channels and the additional channels, mentioned above. The analysis showed that:

- G and PC2 have high correlation with other bands
- NIR-R-PC1 is the best combination based on the plot analysis
- NIR band shows good separability based on the divergence analysis
- PC1-NDVI-S combination shows best separability over three-band combinations based on the divergence analysis.

Therefore, the combination NIR-R-PC1-NDVI -S was selected for classification. The maximum likelihood classification method was used. As expected from their low values in the divergence matrix, grass and trees, airport buildings and residential houses, airport corridors and bare ground, airport buildings and bare ground could not be separated. Using the height information from nDSM, airport ground and bare ground and roads were fused into "ground" and airport buildings with residential houses into "buildings", while trees and grass, as well as buildings and ground could be separated. The final classification is shown in Figure 2. 84% of the building class is correctly classified, while All of 109 buildings have been detected but not fully, the omission error is 9% . Aircrafts and vehicles are again mixed with buildings.

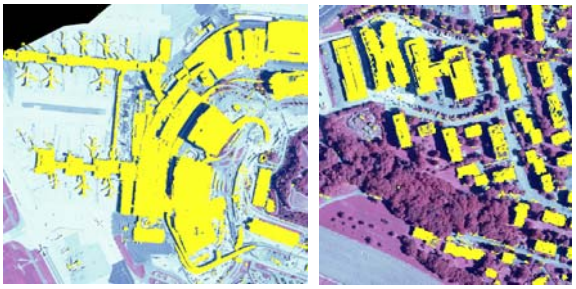


Figure 2. Building detection result from method 2. (Left: airport buildings, Right: residential area).

4.3 Building detection using density of raw Lidar DTM and NDVI (Method 3)

Buildings and other objects, like high or dense trees, vehicles, aircrafts, etc. are characterized by null or very low density in the DTM point cloud. Using the vegetation class from NDVI channel as a mask, the areas covered by trees are eliminated, while small objects (aircrafts, vehicles) are eliminated by deleting them, if their area is smaller than $25m^2$. Thus, only buildings remain (Figure 3). 85% of building class pixels are correctly classified, while 108 of 109 buildings have been detected but not fully extracted, the omission error is 8% .

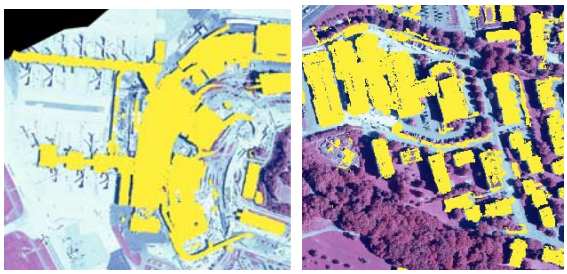


Figure 3. Building detection result from method 3. (Left: airport buildings, Right: residential area).

4.4 Building and tree detection from Lidar data (Method 4)

As mentioned above, in the raw DSM data the point density is generally much higher at trees than at open terrain or buildings. On the other hand, tree areas have low horizontal point density in the raw DTM data. We start from regions that are voids or have low density in the raw DTM (see Method 3). These regions represent mainly buildings and trees and are used as mask to select the raw DSM points for further analysis. In the next step, we used a search window over the raw Lidar DSM data with a size of 5 m x 5 m. Neighboring windows have an overlap of 50%. The window size has a relation with the number of points in the window and the number of the points in the search window affects the quality of the detection result. The method uses all points in the window and labels them as tree if all parameters below have been met. The size of $25m^2$ has been agreed to be enough to extract one single tree. A bigger size may result in wrong detection especially in areas where the buildings are neighboring with single trees. On the other hand, the data has low point density: 1 pt / 2 m^2, that means about 13 pts / 25 m^2. A smaller size will contain less points and this may not be enough for the detection.

The points in each search window are projected onto the xz and yz planes and divided for each projection in eight equal sub-regions using x_{min}, x_{mid}, x_{max}, z_{min} z_{mid1} z_{mid2} z_{mid3} z_{max} as boundary values of sub-regions, with $x_{mid} = x_{min} + 2.5m$, $x_{max} = x_{mid} + 2.5m$, $z_{mid1}=z_{min}+(z_{max}-z_{min})/4$, $z_{mid2} =z_{min}+2*(z_{max}-z_{min})/4$, $z_{mid3}=z_{min}+3*(z_{max}-z_{min})/4$ and similarly for the yz projection. The density in the eight sub-regions is computed. The first step is the detection of trees and the second the subtraction of tree points from all off-terrain points. The trees have been extracted by four different parameters. The parameters have been calculated using tree-masked areas of the raw Lidar DSM data. The tree mask has been generated by Method 2. Then, the calculated parameters (the average of all search windows) have been applied to the raw Lidar DSM data for detection of trees.

The first parameter (s) is similarity of surface normal vectors. We assume that the tree points would not fit to a plane. With selection of three random points in the search window, the surface normal vectors have been calculated n (number of points in search window) times. Then, all calculated vectors have been compared among each other. In case of similar value of compared vectors, the similarity value was increased by adding 1. In the tree masked points, the parameter (s) has been calculated as smaller than 2. The second parameter (vd) is the number of the eight sub-regions which contain at least one point. The trees have high Lidar point density vertically. Thus, at trees more sub-regions contain Lidar points. Using the tree mask, we have observed that at least 5 out of the 8 sub-regions contain points. Thus, the parameter (vd) has been selected as

vd>4. The third parameter (z) is the tree height. Using the tree mask from multispectral classification, we calculated the minimum tree height as 3m. The fourth parameter (d) is the point density. The minimum point density has been calculated for the tree masked areas as 20points/ 25m^2. By applying these four parameters to the raw DSM Lidar data, the tree points have been extracted and eliminated from all off-terrain points to extract the buildings. The workflow can be seen in Figure 4.

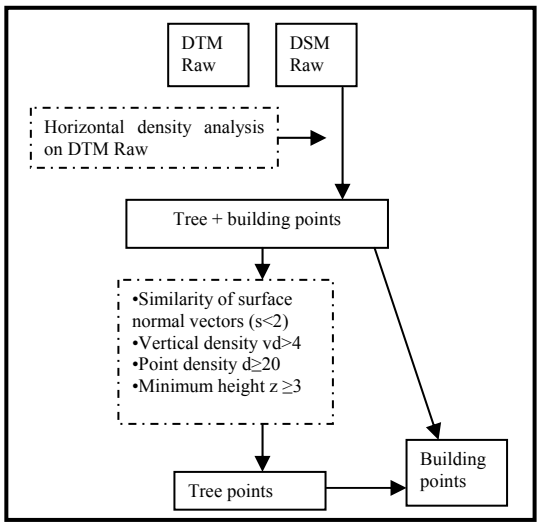


Figure 4. Workflow of detection of buildings in method 4

The density of point cloud directly affects the quality of the result. In addition, some tree areas could not be extracted because of the low point density of the Lidar data. The accuracy analysis shows that 84% of buildings area are correctly extracted, while 100 of 109 buildings have been detected but not fully extracted, the omission error is 17% .(Figure 5).

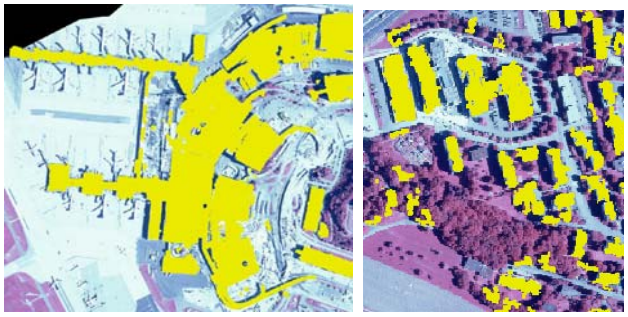


Figure 5. Building detection result from method 4. (Left: airport buildings, Right: residential area).

5. ANALYSIS OF RESULTS

Each method shows similar performance with differences in completeness. The reasons of the failures for correctness and completeness of each method can be seen in Table 2. The improvement of the results is performed by taking into account the advantages and disadvantages of the methods.

	Correctness Failure Reasons	Completeness Failure Reasons
M1	Airplanes/Other moving objects /shadow on vegetation/construction process	Vegetation on roofs, lack of some parts of buildings which are being constructed.
M2	Airplanes/Other moving objects/construction process	Vegetation on roofs, shadow on roofs, lack of some parts of buildings being constructed.
M3	Moving objects (esp. car series in parking lots)/ other man-made structures (highways etc.)	Vegetation on roofs, temporal difference with reference data
M4	Tree groups which could not be extracted and eliminated	Non-detection of small buildings (problem related to low point density), detection of walls as vegetation, temporal difference with reference data

Table 2. The reasons of the failures regarding correctness and completeness for each method (M: Method).

Regarding completeness, the reference data has been generated using aerial images, and some buildings are in construction process. Reference data has been provided from Unique Company and they have produced it using aerial images. But, in the construction areas, these buildings were measured as fully completed, although they were only partly constructed in reality. This increases the omission error especially for the results of the methods 1 and 2 which use aerial images. On the other hand, due to the temporal difference between the reference vector and Lidar data, the completeness of Lidar-based methods (methods 3 and 4) has also been negatively affected.

5.1. Combination of the methods

The results from each method have been combined according to their failures for different types of objects. Intersection of all methods gives the best correctness, while the union of the methods gives the best completeness. The combination of the results has been performed for achieving the best correctness with the best completeness.

(1∩2): While method 2 does not include the errors resulted by the shadow on vegetation, the intersection of these two methods eliminates the problem of shadow on-vegetation (in Figure 12, R1). The correctness of extracted buildings from this combination is 86%, and the omission error is 12%.

(1∩2) ∩4: This combination eliminates the airplane objects from the detection result (Figure 6). Consequently, another advantage of this combination is that it reduces the omission errors which arise from the construction process on some buildings, i.e. multitemporal differences. The correctness of extracted buildings from this result is 96%, and the omission error is 20% (in Figure 12, R2).

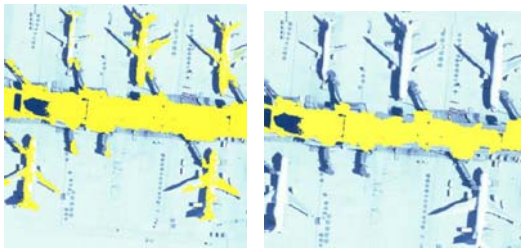


Figure 6. Left: Airplanes which were detected as buildings in (1∩2), Right: Elimination of airplanes with (1∩2) ∩4.

((1∩2) ∩4) ∪ 3: Shadow regions on buildings are replaced with building regions and by this combination (Figure 7). Since method 3 brings the buildings which could not be detected well by method 4, and method 3 is not influenced by shadow, this combination provides better completeness (in Figure 12, R3).

Figure 7. Left: buildings without the regions which covered by shadow in ((1∩2) ∩4), Right: more complete roofs with ((1∩2) ∩4) U 3.

After the union process with the results of the Method-3, the vegetation on the roof tops is still a problem. Intersection of the nDSM and the NDVI algorithms provides the tree and vegetation regions on the roof tops. Intersection of the extracted vegetated regions with building polygons of the Method-4 results in the roof regions which contain vegetation (Figure 8).

Figure 8. Roof regions which contain vegetation.

After adding the roof regions which contain vegetation into the detection result (in Figure 12, R4), the correctness and completeness values are 85% and 7%. As mentioned before, since method 2 have detected all buildings although not fully, the final building polygons should overlap the results from method 2. If the building polygons of result (R4) do not overlap with the results of method 2, they are eliminated. The correctness of the results is improved to 91% and the omission is 7% (Figure 12, R5).

5.2. Using edge information for improvement of correctness

Image data provide edge information, and this can be used to find the precise outlines of the buildings. Firstly, the Canny edge detector (Canny, 1986) has been applied on the orthoimages. The edges have been split into straight lines using corner points which were detected by corner detection (Harris and Stephens, 1988). This has been performed using the Gandalf image processing library (Gandalf, 2009). The straight lines which are smaller than 1 m. have been considered as noise and they have been deleted. The straight lines which may belong to building outlines have been selected using the outline of the detection result (which comes from the combination of methods) and a 2m buffer zone (1m inside, 1 m outside of the building outline). If the straight lines are neighbours in the buffer zone, the longest straight line has been selected. There is an exception for this neighboring criterion: the start or end point of a straight line should not be the closest point to the neighbouring line. With this exception, we avoid the elimination of lines, which are almost collinear (Figure 9).

Figure 9. Left: the straight lines which may belong to the building outline (yellow) and Right: long lines (red).

After selection of the straight lines, they have been converted to closed polygons. For the conversion to polygons, a sorting of the lines in clock-wise direction is used. To perform sorting, the travelling salesman convex hull algorithm (Deineko et al., 1992) has been applied. After closing the polygons, we separate the lines into those that were detected from the images (red) and the ones added by this algorithm (blue) (see Figure 10). The red straight lines, which are shorter than 10 m. and form an acute angle (between 1 and 80 degrees), are eliminated (Figure 10), as well as all blue lines.

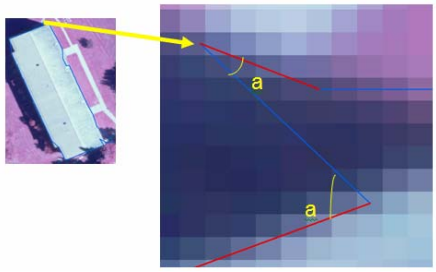


Figure 10. Line elimination procedure when the line length is shorter than 10 meters and has acute angle with its neighbouring lines (red: eliminated lines, blue: lines added by the travelling salesman algorithm, yellow: acute angle).

If two red lines form an acute angle and are shorter than 10 m., then both lines are eliminated. After this elimination, the travelling salesman convex hull algorithm has been applied again using the non-eliminated red lines and generated the refined building polygons (Figure 11).

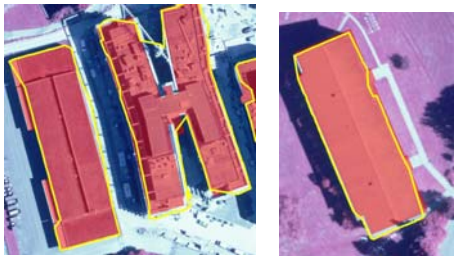


Figure 11. Final building polygons (yellow)., and reference data (red).

After this process, the correctness has been improved to 94% with remaining 7% omission error (Figure 12, R6). However, it has not been applied on all 109 buildings of this test yet, due to time restrictions, while it has shortcomings, as the travelling salesman algorithm does not use any input data information for forming closed polygons.

5.3 Final results

The rule-based system for the combination of methods can be seen in Figure 12.

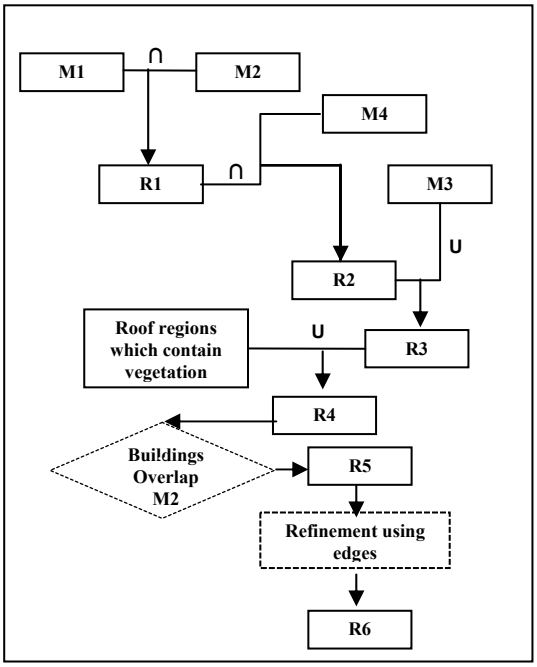


Figure 12. Combination of the methods. R: result from combination, M: Method**.**

Table 3 gives a summary of the correctness and omission percentages of the various detection methods.

	Correctness (%)	Omission (%)		Correctness (%)	Omission (%)
M1	83	7	R1	86	12
M2	84	9	R2	96	20
M3	85	8	R3	85	8
M4	84	17	R4	85	7
			R5	91	7
			R6	94	7

Table 3. Summary of the correctness and omission percentages.

6. CONCLUSIONS

In this paper, different methods for object detection (mainly buildings) in Lidar data and aerial images have been presented. In each method, the basic idea was to get first preliminary results and improve them later using the results of the other methods. The methods have been tested on a dataset located at Zurich Airport, Switzerland, containing RGB and CIR, Lidar DTM and DSM point clouds and regular grids and building vector data for accuracy assessment. The results from each method have been combined according to their error characteristics. Edges have been used for further improvement of the detected building outlines. Finally, the correctness of detection has been 94% with remaining 7% omission error that mostly comes from construction process on airport buildings. Future work will focus on the improvement of use of edges, using the Lidar DSM to eliminate lines which don't belong to buildings and 3D building roof modeling.

ACKNOWLEDGEMENTS

This work has been supported by the EU FP6 project Pegase. We acknowledge data provided by Swisstopo and Unique Company (Airport Zurich).

REFERENCES

Axelsson, P., 2001. Ground estimation of laser data using adaptive TIN-models. Proc. of OEEPE workshop on airborne laserscanning and interferometric SAR for detailed digital elevation models, 1-3 March, Stockholm, Sweden, pp. 185-208.

Brenner, C., 2000. Towards fully automatic generation of city models. Int. Archives of Photogrammetry and Remote Sensing, Vol 33, Part B3/1, pp.85-92.

Brovelli, M.A., Cannata, M., Longoni U.M., 2002. Managing and processing Lidar data within GRASS. Proc. of the GRASS Users Conf. 2002, Trento, Italy. http://citeseer.ist.psu.edu/541369.html (accessed 02 July 2009).

Canny, J., 1986. A computational approach to edge detection. IEEE Trans. Pattern Anal. Machine Intell., 8(6), 679–698.

Deineko, V., Van Dal, R., Rote, G., 1992. The convex-hull-and-line traveling salesman problem: A solvable case. Information Processing Letters, 51 (3), 141–148. http://citeseer.ist.psu.edu/old/286488.html (accessed 02 July 2009).

Durupt, M., Taillandier, F., 2006. Automatic building reconstruction from a digital elevation model and cadastral data : An operational approach. IAPRS , Vol. 36, Part 3, pp. 142-147. http://www.isprs.org/commission3/proceedings06/singlepapers/O_14.pdf (accessed 02 July 2009).

Elberink, S.,O., Vosselman, G., 2006, 3D Modelling of Topographic Objects by Fusing 2D Maps and LIDAR Data, IAPRS* Vol. 36, Part 4, pp. 199-204. http://intranet.itc.nl/papers/2006/conf/vosselman_3D.pdf (accessed 02 July 2009).

Elmqvist, M., Jungrt, E., Lantz, F., Persson, A., Soderman, U., 2001. Terrain modelling and analysis using laser scanner data. IAPRS*, Vol. 34, Part 3/W4, pp. 219-227. http://www.isprs.org/commission3/annapolis/pdf/Elmqvist.pdf (accessed 02 July 2009).

Gandalf, 2009. http://gandalf-library.sourceforge.net/ (accessed 27 June 2009).

Gruen, A., Wang, X., 1998. CC-Modeler: A topology generator for 3-D city models. ISPRS Journal of Photogrammetry & Remote Sensing 53(5), 286-295. http://linkinghub.elsevier.com/retrieve/pii/S0924271698000112 (accessed 02 July 2009).

Haala, N., and Brenner, C., 1999. Virtual city models from Laser altimeter and 2D map data. Photogrammetric Engineering & Remote Sensing 65 (7), 787-795. http://www.ifp.uni-stuttgart.de/publications/1999/norbert_ohio.pdf (accessed 02 July 2009).

DENSE MATCHING IN HIGH RESOLUTION OBLIQUE AIRBORNE IMAGES

M. Gerke

International Institute for Geo-Information Science and Earth Observation – ITC, Department of Earth Observation Science, Hengelosestraat 99, P.O. Box 6, 7500AA Enschede, The Netherlands, gerke@itc.nl

KEY WORDS: Adjustment, Bundle, Calibration, Matching, Point Cloud, Rectification

ABSTRACT:

An increasing number of airborne image acquisition systems being equipped with multiple small- or medium size frame cameras are operational. The cameras normally cover different viewing directions. In contrast to vertical images, those oblique images have some specific properties, like a significantly varying image scale, and more occlusion through high raising objects, like buildings. However, the faces of buildings and other vertically extended objects are well visible and this is why oblique images are used for instance for visualization purposes.
This paper shows results from applying the sophisticated Semi-Global-Matching technique to a set of oblique airborne images. The images were acquired by two systems, namely FLI-MAP 400 (Fugro Aerial Mapping B.V.) and Pictometry (BLOM Aerofilms) over the same area. After the joint adjustment of the images, dense matching and forward ray intersection was performed in several image combinations. The disparity maps were evaluated through the comparison with a reference map derived from LIDAR which was acquired in parallel with the FLI-MAP system. Moreover, the 3D point clouds were analyzed visually and also compared to the reference point cloud. Around 60 to 70 percent of all matches were within a range of $\pm$ 3pix to the reference. Since the images were acquired in different flight configurations, the impact of different intersection angles and baselines to the triangulation is quite obvious. In general, the overall structures on the building faces are well represented, but the noise reduction needs further attention.

1 INTRODUCTION

An increasing number of airborne image acquisition systems are operational (Petrie and Walker, 2007). Because of the availability of low-cost digital cameras with small or medium sized sensors, some of those systems carry multiple cameras covering different viewing directions. For instance from Pictometry[1] image are available already for a number of cities and they are accessible in the category "birds eye view" in Microsoft Bing Maps[2] (formerly known as Virtual Earth).

The use of oblique images for topographic mapping purposes was shown in quite some papers. In (Höhle, 2008) height determination from single oblique images is demonstrated. The verification of vector data using oblique imagery is shown in (Mishra et al., 2008). Due to the fact that building façades are well visible in oblique images, some researchers concentrate on how to automatically extract façade textures (Früh et al., 2004, Wang et al., 2008). Besides, the oblique images are interesting for cadastre applications, because the building outline as defined at the vertical wall is directly visible (Lemmen et al., 2007). Compared to vertical airborne images, oblique images have some specific properties. Depending on the tilt angle, the scale within the imaged scene varies considerably. Moreover, vertical structures of raised objects like buildings or trees are imaged, but the (self)occlusion by those objects is much more significant compared to the vertical image case.

Another interesting application and research domain concerns the derivation of high dense point information through image matching techniques. The benchmark results from the Middlebury[3] testsets show that high quality state-of-the-art techniques to dense matching are available. If it is possible to apply those techniques to oblique airborne images, interesting new applications would arise, or support existing ones, like the ones listed above. In general, point clouds as derived from dense matching in oblique images can be a complementary data source to airborne laser scanning, as those devices normally do not capture dense points on vertical structures. Of course, the traditional use of this kind of data to produce digital surface or terrain models is another possible application.

In (Besnerais et al., 2008) an approach to dense matching in oblique airborne images is presented. The authors develop a pixel wise similarity criterion which accounts for the special viewing geometry of oblique images. A dense depth map is obtained through global regularization. The approach was tested on a number of test images and showed good results. However, the ground sampling distance of the used images was not smaller than 1.4m, mostly it was even larger, up to 20m.

This paper evaluates the application of the Semi-Global-Matching technique (SGM, see (Hirschmüller, 2008)) to a set of high resolution FLI-MAP[4] and Pictometry images. One particular façade of a building is normally only visible in images taken from one viewing direction, resulting in a relatively bad intersection angle in object space. Thus, the main objective of this paper is to evaluate the overall accuracy of the derived 3D point cloud as derived from a forward intersection of matched points. Although other – may be better performing– algorithms for dense matching exist (Seitz et al., 2006) we chose SGM, because it demonstrated already its fitness for the photogrammetric production process, c.f. (Hirschmüller et al., 2005).

As no sufficient calibration and orientation information was available, the whole block first needed to be adjusted. The method for bundle block adjustment, including self-calibration of multiple devices and employing scene constraints to enhance the scene geometry was introduced and tested in (Gerke and Nyaruhuma, 2009). The dense matching algorithm then was applied to several combinations of stereo images, and results were evaluated through LIDAR data which was acquired from the FLI-MAP system.

[1] http://www.pictometry.com
[2] http://www.bing.com/maps
[3] http://vision.middlebury.edu/stereo/ (accessed 15 March 2009)
[4] http://www.flimap.nl

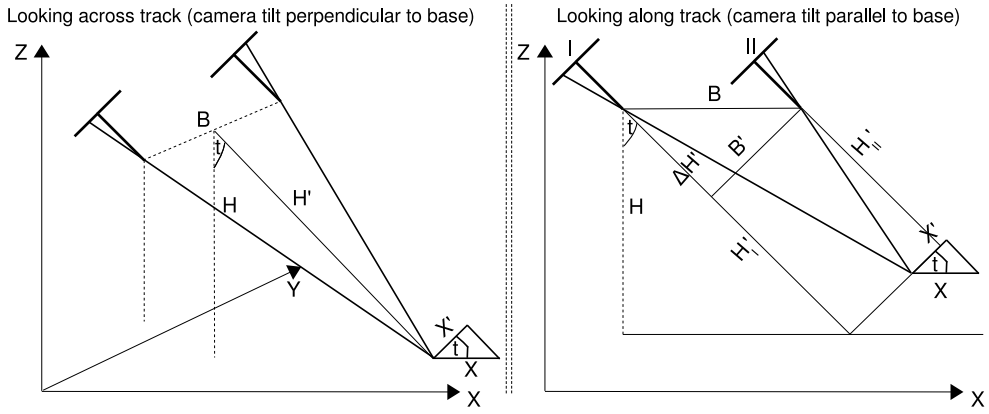


Figure 1: Geometry of across track (left) and along track (right) baseline in oblique stereo images

2 STEREO GEOMETRY AND ACCURACY IN OBLIQUE IMAGES

The dense matching results in point correspondences in a stereo pair[5]. The image rays as resulting from those matching image pairs are forward intersected in order to compute 3D points in object space. In this section a brief theoretic approximatoin of the expected accuracy from the given intersection geometry in oblique airborne images is derived. Two cases can be distinguished: A) the two cameras used for the stereo intersection are oriented across track, i.e. they inclose a right angle with the base, and B) the cameras are looking in flight direction. The intersection geometry of the first case can be derived from the standard normal case, but the varying scale and the tilt of the camera coordinate system wrt the actual coordinate system need consideration. The second case can be compared with the longitudinal tilt-setup (Albertz and Kreiling, 1980), i.e. the normal case with different camera heights. Only here the whole base is tilted. In Fig. 1 both camera geometries are sketched.

The scale within an oblique image depends on the flying height H, the focal length c the tilt angle t, and the angle between the viewing ray to a target and the vertical β (symbols according to (Höhle, 2008)):

$$m = \frac{H \cdot \cos(\beta - t)}{c \cdot \cos\beta}, \tag{1}$$

where m: scale at target point. At the principal point β equals t, whereas at the fore- and the background β is determined from t and the half field of view α: $\beta_{fore} = t - \alpha$ and $\beta_{back} = t + \alpha$.

A: tilt across track (side-looking) In the vertical image case, the accuracy for triangulated points in height (s'_H), and in X-Y plane ($s'_{X,Y}$) can be estimated by:

$$s'_H \approx \frac{H'}{B} \cdot m \cdot s_{px}, \tag{2}$$

$$s'_{X,Y} \approx s_x \cdot m \approx s_y \cdot m, \tag{3}$$

where $s_x \approx s_y \approx 0.5 \cdot s_{px}$ are the standard deviations for image coordinate and parallax measurements; the errors in the orientation components are neglected. In the case of the tilted camera system, these formulas are applicable to the tilted system, so the varying scale needs to be considered, according to equation 1, also H' needs to be adopted accordingly:

$$H' = m \cdot c. \tag{4}$$

[5]For the combination of multiple views see the experiment section

Finally, the respective error components need to be projected from the tilted system to the actual coordinate system:

$$s_H \approx \sqrt{(s'_H \cdot \cos t)^2 + (s'_{X,Y} \cdot \sin t)^2}, \tag{5}$$

$$s_{X,Y} \approx \sqrt{(s'_H \cdot \sin t)^2 + (s'_{X,Y} \cdot \cos t)^2}, \tag{6}$$

thus for a tilt angle of 45° both components will be identical.

B: tilt along track (forward-looking) To derive the accuracy in the tilted system H, X', Y', first the necessary parameters for the case of longitudinal tilt need to be computed: Base B' in the tilted system and the heights of the cameras I and II:

$$B' = B \cdot \cos t, \tag{7}$$

$$\Delta H' = B \cdot \sin t, \tag{8}$$

$$H'_I = m \cdot c, \quad \text{and} \quad H'_{II} = H'_I - \Delta H'. \tag{9}$$

Applying partial derivation wrt the image and parallax measurements to the formulas given in (Albertz and Kreiling, 1980), the accuracies for the coordinate components in the tilted system can be derived:

$$s'_{H_I} \approx s'_{H_{II}} \approx \sqrt{\left(\tfrac{H_{II}'}{p_x}\right)^2 \cdot s_{px}^2 + \left(\tfrac{B' \cdot \sin t}{p_x}\right)^2 \cdot s_x^2}, \tag{10}$$

$$s'_{X,Y} \approx \tfrac{H_I'}{c} \cdot s_x. \tag{11}$$

Note that the actual parallax needs to be computed for the estimation. In the approximations for the given data, see below, a mean parallax according to a mean depth in fore- and background was assumed. For the planar accuracy the more pessimistic estimation, assuming the smaller image scale, is given here. Finally, the planar and height components in the actual coordinate system are computed according to equations 5 and 6.

3 METHODS ADOPTED

3.1 Block adjustment for multiple platforms

In (Gerke and Nyaruhuma, 2009) a method to incorporate scene constraints into the bundle block adjustment is described and tested. The bundle block adjustment algorithm uses horizontal and vertical line features, as well as right angles to support the stability of block geometry. Those features can be identified at building façades, as visible in oblique images. In addition, the approach is able to perform self-calibration on all devices which are incorporated in the block. This is an important issue in the case of oblique images, as those are often acquired by non-metric cameras. The extension to the setup used for this paper where images from different platforms are involved is done without any change to the core approach.

Parameter	FLI-MAP vertical	FLI-MAP oblique	Pictometry oblique
flying height [m]	275	275	920
baseline [m]	50	50	400
tilt angle [°]	0	45	50
number of images,viewing direction	7	8xSW	2xW, 2xS, 2xE, 1xN
focal length [mm]	35	105	85
pixel size [μm]	9	9	9
sensor size [mm x mm]	36x24	36x24	36x24
GSD and theoretic accuracies (for oblique: from fore- to background)			
ground sampling distance [cm]	7	2.8 – 4	10 – 16
$s_{X,Y}$, vertical [cm]	4	NA	NA
s_Z, vertical [cm]	40	NA	NA
$s_{X,Y}$, across track base [cm]	NA	NA	22 – 44
s_Z, across track base [cm]	NA	NA	18 – 37
$s_{X,Y}$, along track base [cm]	NA	60 – 92	22 – 42(*)
s_Z, along track base [cm]	NA	60 – 92	19 – 35(*)

(*): along track base images from Pictometry were not used.

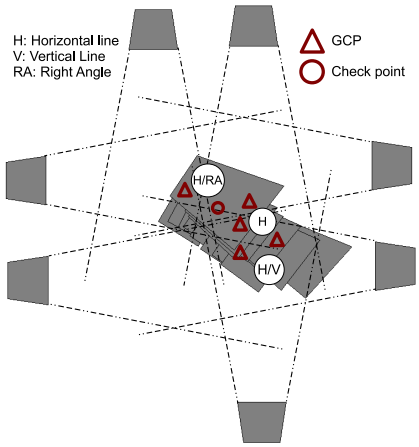


Figure 2: Image parameters and layout of sample block

3.2 Rectification and dense matching

The approach to dense stereo matching as applied in the current implementation is the Semi-Global-Matching algorithm (Hirschmüller, 2008). The basic idea behind this technique is to aggregate local matching costs by a global energy function, which is approximated by an efficient pathwise 1-dimensional optimization.

To simplify the matching, the images are rectified beforehand. For this purpose the approach proposed in (Oram, 2001) is applied. A homography is estimated which is compatible to the fundamental matrix (Hartley and Zisserman, 2004, chap. 13). The aim is to minimize distortions due to perspective effects, and thus also to reduce the disparity search space. A consequence from the particular algorithm is that the epipolar lines are coincident, but not necessarily parallel. Hence, in the subsequent rectification the images need resampling to obtain parallel epipolar lines. One disadvantage of this procedure is that straight lines are not preserved, however, this does not influence the matching. To compute the optimal homography point correspondences are required, like for instance in the case at hand the adjusted tie points. If images are taken from approximately the same viewing direction, it is also possible to extract further matches through scale invariant point descriptors like SIFT (Lowe, 2004). Outliers in the correspondences are identified through RANSAC within the estimation of the compatible homography. The inliers are here also used to estimate the disparity search range for the dense matching.

4 EXPERIMENTS

4.1 Description of used data

Part of the data used for these experiments was acquired by the Fugro Inpark FLI-MAP 400 system in March 2007 over Enschede, The Netherlands. Besides two LIDAR devices and two video cameras, the system carries two small-frame cameras, one pointing vertical, and one oblique camera, looking in flight direction, tilted by approx. 45°. Additional Pictometry images were made available through BLOM Aerofilms. Those images were acquired only one month before the FLI-MAP data. A small block of 7 vertical and 8 oblique images from FLI-MAP as well as 7 images from Pictometry was chosen for the experiments. In Fig. 2, upper part some parameters of the images are given, the GSD and accuray estimation was done according to equations 1 to 11, while a standard deviation for image measurements of a half pixel was assumed. In the bottom of that figure the layout of the block is shown, including GCP, check points and the approximate position of defined scene constraints. The highly overlapping images in the center are from the FLI-MAP acquisition, while the 7 regularly aligned outer images are from the Pictometry-flight. Note that no along track images are chosen from Pictometry. The airplane acquired the images in N-S-direction, so the East- and West-looking images belong to one flight line (baseline approx. 400m) and the two South-looking images are from two adjacent strips, baseline approx. 350m. For the accuracy estimation the two South-looking images can be treated like across-track images.

4.2 Block adjustment results

Four full and one height GCP were used for the adjustment. Additionally, one right angle, 3 horizontal and 4 vertical line constraints were defined. It was assured that in every image at least one of the features used for the scene constraints was visible. In Table 1 the adjustment results in terms of RMSE at the control and check points, or features respectively are listed. One observation from the residuals is that the Z-component is smaller than the X/Y values for all control and check features. Also the residuals at vertical constraints are larger than the residuals at horizontal constraints, and those are also influenced by the Z-component only. One reason for this can be that the tilt of the Pictometry images is larger than 45° and thus the X,Y-component is less accurate than the Z-component, refer also the the listed theoretic accuracies in Fig. 2. One general drawback of this block-setup is that outside the overlapping areas no GCPs or scene constraints are available and applicable, respectively, so the overall block geometry at the borders is not optimal. However, since the residuals at the façades are at least for the Pictometry images less than one pixel this result can be considered satisfactory.

Assessment	RMSE value[cm]
X-Res. at GCP	2.1
Y-Res. at GCP	4.8
Z-Res. at GCP	1.3
X-Res. at Check	16.2
Y-Res. at Check	5.8
Z-Res. at Check	1.5
Res. at H-constraints	1.4
Res. at V-constraints	6.8
Res. at RA-constraints (°)	0.01

Table 1: Residuals from bundle block adjustment

4.3 Dense matching results

The dense matching was performed in several stereo image combinations. Besides the matching in images from one platform,

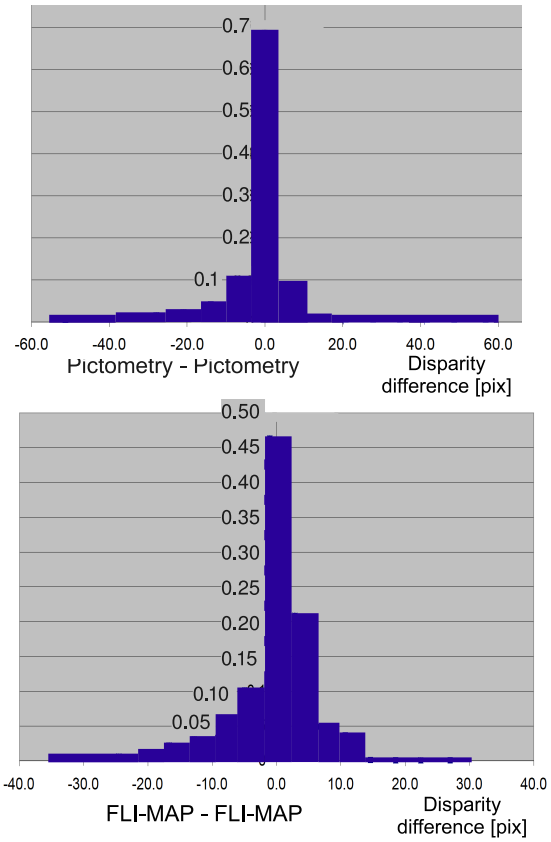


Figure 3: Two sample relative histograms of disparity differences wrt reference disparity map. Note the scale difference between both diagrams.

matching between FLI-MAP and Pictometry was tested. This is interesting, because by this means the scene can be observed from approximately the same direction through multiple views. The overlap from consecutive Pictometry images is not large enough to create 3-ray points, however, incorporating also FLI-MAP images makes this possible. Besides, this setup gives an interesting geometry for forward intersection.

Two methods were used to assess the results: one quantitative and one qualitative. For the quantitative assessment a reference disparity map was computed from the FLI-MAP LIDAR data, then the differences to the disparities from image matching were analyzed using histograms. For a more qualitative assessment 3D point clouds were computed from the matching results and then assessed visually, also in comparison to the LIDAR point cloud.

Disparity map assessment For this assessment the reference LIDAR points (density: 20 points per m^2) were projected into the image plane as defined by the respective image orientation and calibration parameters and subsequently a reference disparity map was computed. Two issues are important here: first, only first pulse LIDAR points should be considered, as also in the image only the visible surface can be matched. Second, through the oblique viewing direction as realized with the cameras one has to take into account self-occlusion through buildings; the laser scanner scans vertical and thus scans other parts of the scene, especially on the backside of buildings visible in the images. To avoid errors from that circumstance, only areas which do not show these effects were used for the evaluation.

The disparity maps were assessed by calculating the difference disparity map and computing a histogram out of that one. Only pixels showing a disparity value in both maps were considered,

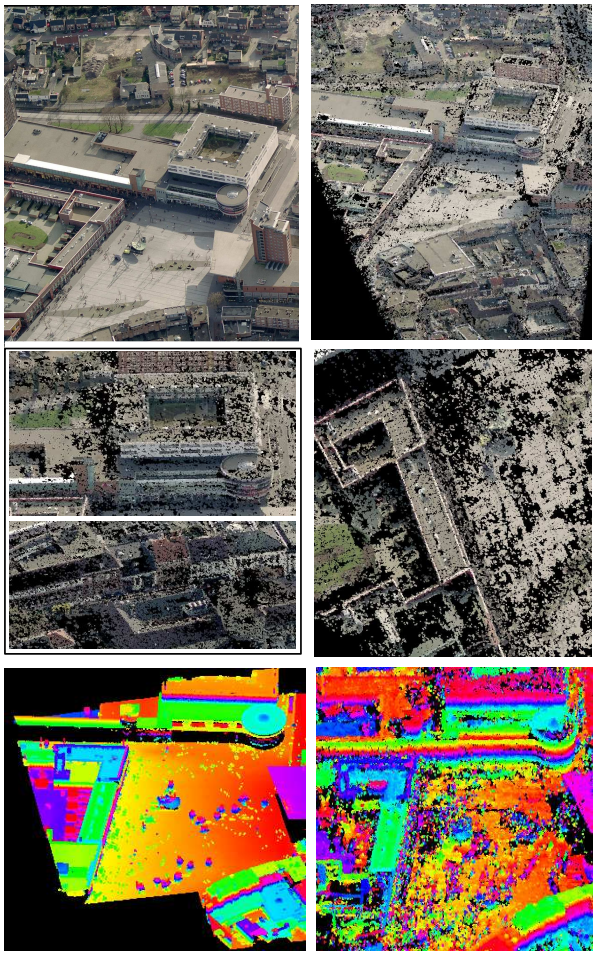


Figure 4: Results from dense matching in two overlapping South-looking Pictometry images. Top: left image and 3D cloud from matching, center row: zoom to point cloud from matching at façades (left) and top view (right), bottom row: point cloud color coded height: reference (left), from matching (right)

thus matched points at façades which were not acquired by the LIDAR device can not be assessed. Two of such relative histograms are shown in Fig. 3. The upper histogram shows the differences from the matching within two Pictometry images (see Fig. 4). For this histogram approx. $50 \cdot 10^3$ matches were considered (out of $2.2 \cdot 10^6$ in total), and around 70% of them show a difference of ± 3 pixels to the reference. The histogram at the bottom shows the analysis from the matches within two oblique images from FLI-MAP, refer to Fig. 5. For this histogram approx. $200 \cdot 10^3$ matches were considered (out of $6.4 \cdot 10^6$ in total). Because of the smaller baseline between consecutive FLI-MAP images, compared to Pictometry, the overlapping area is larger, and thus results in more matches. Approximately 60% are within the difference of ± 3 pixels. All matches outside this tolerance can be considered as blunder. A more in depth analysis revealed that most blunders were caused in shadow areas or other areas with poor texture. When assessing those histograms is should be considered that errors from the image calibration and post estimation also contribute to those residuals, thus a final conclusion on the absolute matching accuracy of the SGM implementation can not be made.

Point clouds: Pictometry to Pictometry For the following evaluations a forward intersection of the matched points was performed. A simple blunder detection was implemented by applying a threshold to the residual for image observations. For two-ray intersections this method can filter some blunders, but be-

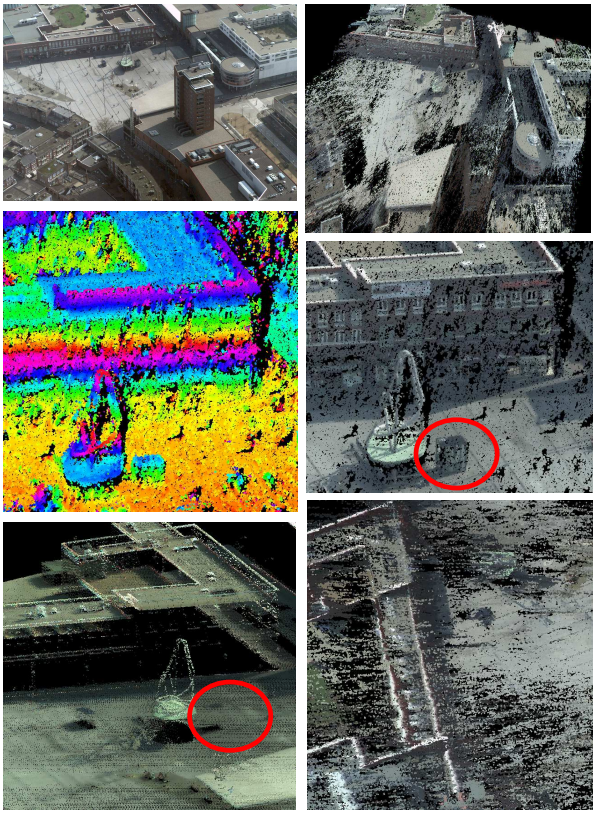


Figure 5: Results from dense matching in two overlapping FLI-MAP images. Top: part of left image and point cloud from matching, centre: 3D point cloud from matching, color coded height (left) and color picked from images, bottom row: reference point cloud and top view to matched point cloud . The circle indicates an elevator box which is visible in the point cloud from matching, but not in the laser scanning data.

cause only one redundant observation is available, quite a lot of blunders will not be detected.

The point cloud as resulted from the triangulation of matches in a Pictometry image pair are shown in Fig. 4. The top row in that figure shows part of the left image and an overview on the 3D scene defined by the matched point cloud. The center row shows a zoom in to the colored point cloud from matching, focusing on some façades and the vertical view to that scene. Finally, the bottom row shows the reference point cloud at the left hand side, where the color codes the height (one full color cycle equals 10m). The corresponding part of the matched point cloud is depicted on the right hand side. From that figure some interesting observations can be made. Although most of the flat roofs show poor texture, the corresponding part in the matched point cloud is quite dense and homogeneous. However, the height of the roof in the upper part is not correct, it is approx. 50cm lower than shown in the reference point cloud. In the vertical view on the point cloud from SGM the occluded areas are clearly visible, whereas the vertical façades are not visible in the reference point cloud. Overall, the structures are well represented, but the mismatched pixels impose a certain level of noise to the scene. Those mismatches can hardly be detected, as only stereo matches are available in this case. The detailed zoom on the vertical image shows that the accuracy in x-y direction is quite good, and apparently even better than the estimated one ($s_{X,Y}$: 22 – 44cm).

Point clouds: FLI-MAP to FLI-MAP The triangulated point cloud from the matching in two consecutive oblique images from the FLI-MAP data is shown in Fig. 5. The zoom in to the colored point cloud shows quite some details, for instance the elevator box close to the statue is clearly visible. The statue itself is well represented and even gaps in the matches where people were walking during image exposure are clearly identifiable. However, those views onto the point cloud were made from the viewing direction of the cameras, so the main error showing and effect in viewing direction is not visible. The estimated depth accuracy (in viewing direction, s'_H) of the along-track FLI-MAP data varies from 90 to 130cm, and the error in X', Y'-direction is only 2cm. To assess the overall quality, the vertical view needs to be considered: Here the uncertainty in viewing direction is quite obvious. If the result from the vertical view-zoom is compared to the one from the Pictometry data (Fig.4, center row), it can be observed that the result from the FLI-MAP data is more inaccurate. This visually achieved observation confirms the theoretically approximated accuracy, which is about four times worse.

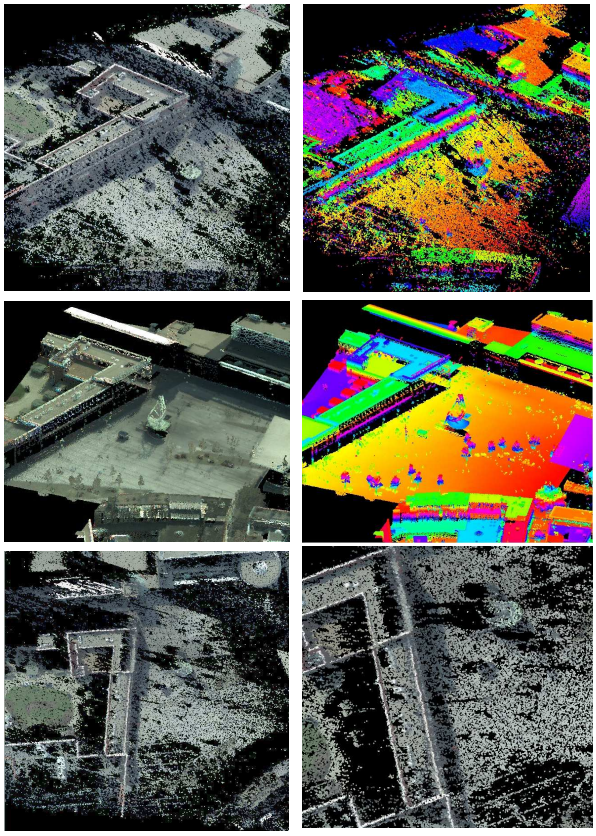


Figure 6: Results from multiple view triangulation. Top: matched point cloud, color and height, center: reference point cloud, bottom: top view

Point clouds: Multiple view For this experiment it was desired to exclude the wrong matches from the triangulation. To achieve this goal, the dense matching results from several matches were combined in the following manner: Dense matching was performed in the three combinations: ① FLI-MAP1 ↔ FLI-MAP2; ② FLI-MAP2 ↔ Pictometry1; ③ FLI-MAP1 ↔ Pictometry1. The matches from ① and ② are linked in a way that the matching points from the right image of ① are searched in the left image of ② and by this means corresponding matches ③' FLI-MAP1 ↔ Pictometry1 are created. In the subsequent triangulation only those matches were considered which coincide with the original matches from ③. Thus it can be expected that through this double check some blunders were removed. For more details on this method see (Gerke, 2008). In Fig. 6 some details on the results are shown. The point cloud contains now less points ($1.6 \cdot 10^6$ points from the FLIMAP-only point cloud vs. $190 \cdot 10^3$

matches here), but the overall accuracy seems to be better, see e.g. the height of the buildings. Also the detailed look from vertical shows less noise, compared to the two-fold matches before.

5 CONCLUSIONS AND OUTLOOK

This paper reports about the utilization of the dense image matching technique Semi-Global-Matching to a set of high resolution oblique airborne images. The images were acquired from different platforms and thus in different configurations. The comparison of the depth maps from matching with a reference computed from LIDAR data showed that roughly 70% of all matches are within an error range of $\pm$ 3pixel, however, also the residual errors from camera calibration and orientation have an impact on this evaluation. The remaining matches can be considered as blunders. How can those blunders be removed and the noise level be reduced? If multiple overlaps are available, sophisticated error analysis prior to triangulation is feasible (Hirschmüller, 2008). Also the method as applied here shows good results, namely to eliminate wrong matches through linking matches of adjacent images and applying a double check through a direct match. Other authors use the much stronger trinocular stereo geometry for matching (Heinrichs et al., 2007), or apply a similarity criterion for multiple views directly (Besnerais et al., 2008). If only two-fold overlap is available – as mostly in facade observations from oblique images – one method could be to incorporate reliable SIFT features within the SGM approach directly: The disparities as defined by them can be used to reduce the respective matching cost.

The point cloud as resulted from the triangulation of the respective matches revealed the sensitivity to the ray intersection angle and base length of cameras. For instance in the case of consecutive FLI-MAP images the theoretic standard deviation of triangulated points in viewing (depth) direction is – due to the small effective baseline – around 1m, but perpendicular to that – due to the low flying height – around 2cm only. In the shown examples these theoretic measures were confirmed. In the tested images from Pictometry the intersection geometry is better because of the longer baseline. In general, the overall structures on the building faces are well represented, but the noise reduction needs further attention.

In the current research the focus is put on the automatic detection and extraction of buildings in oblique images. Here, the point cloud as derived from the matching can give valuable cues. Another issue concerns the derivation of a more complete coverage by merging the point clouds as derived from different viewing directions. At least for the roof areas this can be done in a similar manner as shown above, namely through linking matches, since the majority of roof areas are visible from multiple directions.

ACKNOWLEDGEMENTS

I want to thank Daniel Oram for providing the code for general rectification on his homepage[6]. Further I like to thank Matthias Heinrichs, TU Berlin, for providing me with his code for Semi-Global-Matching. I also want to thank BLOM Aerofilms for providing the Pictometry dataset. Finally I would like to thank the anonymous reviewer for their comments.

REFERENCES

Albertz, J. and Kreiling, W., 1980. Photogrammetrisches Taschenbuch. 3. edn, Herbert Wichmann, Karlsruhe.

Besnerais, G. L., Sanfourche, M. and Champagnat, F., 2008. Dense height map estimation from oblique aerial image sequences. Computer Vision and Image Understanding 109(2), pp. 204 – 225.

Früh, C., Sammon, R. and Zakhor, A., 2004. Automated texture mapping of 3d city models with oblique aerial imagery. In: 3D Data Processing Visualization and Transmission, International Symposium on, pp. 396–403.

Gerke, M., 2008. Dense image matching in airborne video sequences. In: ISPRS: XXI congress : Silk road for information from imagery, Vol. XXXVII-B3b, International Archives of Photogrammetry and Remote Sensing, Beijing, pp. 639–644.

Gerke, M. and Nyaruhuma, A., 2009. Incorporating scene constraints into the triangulation of airborne oblique images. In: High-Resolution Earth Imaging for Geospatial Information, Vol. XXXVIII, International Archives of Photogrammetry and Remote Sensing, Hannover.

Hartley, R. I. and Zisserman, A., 2004. Multiple View Geometry in Computer Vision. Second edn, Cambridge University Press.

Heinrichs, M., Hellwich, O. and Rodehorst, V., 2007. Efficient Semi-Global Matching for trinocular stereo. In: IAPRS, Vol. 36, pp. 185–190. Part 3-W49A (PIA conference, Munich, Germany).

Hirschmüller, H., 2008. Stereo processing by Semi-Global Matching and Mutual Information. IEEE Transactions on Pattern Analysis and Machine Intelligence 30(2), pp. 328–341.

Hirschmüller, H., Scholten, F. and Hirzinger, G., 2005. Stereo vision based reconstruction of huge urban areas from an airborne pushbroom camera (HRSC). In: Lecture Notes in Computer Science: Pattern Recognition, Proceedings of the 27th DAGM Symposium, Vol. 3663, Springer, Berlin, pp. 58–66.

Höhle, J., 2008. Photogrammetric measurements in oblique aerial images. Photogrammetrie Fernerkundung Geoinformation 1, pp. 7–14.

Lemmen, M., Lemmen, C. and Wubbe, M., 2007. Pictometry : potentials for land administration. In: Proceedings of the 6th FIG regional conference, International Federation of Surveyors (FIG).

Lowe, D. G., 2004. Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision 60(2), pp. 91–110.

Mishra, P., Ofek, E. and Kimchi, G., 2008. Validation of vector data using oblique images. In: Proceedings of the 16th ACM SIGSPATIAL International conference on advances in Geographic Information Systems, ACM, Irvine, California.

Oram, D., 2001. Rectification for any epipolar geometry. In: Proceedings of the 12th British Machine Vision Conference (BMVC).

Petrie, G. and Walker, A. S., 2007. Airborne digital imaging technology: a new overview. The Photogrammetric Record 22(119), pp. 203–225.

Seitz, S. M., Curless, B., Diebel, J., Scharstein, D. and Szeliski, R., 2006. A comparison and evaluation of multi-view stereo reconstruction algorithms. CVPR 1, pp. 519–526.

Wang, M., Bai, H. and Hu, F., 2008. Automatic texture acquisition for 3d model using oblique aerial images. In: Intelligent Networks and Intelligent Systems, International Workshop on, pp. 495–498.

[6] http://www.aspa29.dsl.pipex.com (accessed 15 March 2009)

COMPARISON OF METHODS FOR AUTOMATED BUILDING EXTRACTION FROM HIGH RESOLUTION IMAGE DATA

G. Vozikis

GEOMET Ltd., Faneromenis 4, 15561 Holargos-Athens, GREECE
george.vozikis@geomet.gr

KEY WORDS: Photogrammetry, Building , Detection , Transformation, Model, Pattern

ABSTRACT:

This paper discusses a comparison analysis of different methods for automated building extraction from aerial and spaceborne imagery. Particularly approaches employing the Hough Transformation, Pattern Recognition Procedures and Texture Analysis are examined. Throughout this investigation advantages and disadvantages of the mentioned methods are examined, in order to see which procedures are suitable for extracting the geometric building properties, and thus to automatically create a DCM (Digital City Model). The examined data sets consist of panchromatic imagery coming from both very high resolution satellites, as well as line scanning aerial sensors. A quantitative and qualitative assessment will help to evaluate the previously mentioned procedures.

1. INTRODUCTION

Automated building extraction from high resolution image data (either airborne or spaceborne) is becoming more and more mature. Everyday new techniques are investigated and the results are getting more and more reliable, while the degree of automation increases. Each building extraction method is of course coupled to certain pros and cons. The use of the Hough Transformation has proven to be a very promising tool in the frame of the automated creation of Digital City Models (DCMs), by extracting building properties from optical data. But also approaches based on Image Matching or Texture Analysis seem to provide usable results. A DCM is described through the outlines of buildings outlines of an urban area. Vertical walls are assumed, and the elevation information of these buildings can be taken from a DSM (Digital Surface Model). The creation of the DSM and the assignement of the elevation value is not discussed in this paper, thus when mentioning DCMs we actually mean the Model that holds the 2D outline-information of a building.

The goal of this paper is to conclude for which kind of data sets and accuracy pretensions a certain approach is recommendable. Moreover, the reachable degree of automation is also examined, in order to see how reliable results are that were produced without human interaction.

Table 1: Examined data sets.

Sensor	Location	GSD (m)	Extents (km)
ADS40	Valladolid, Spain	0.25	1 x 1
HRSC-AX	Bern, Switzerland	ca. 0.3	0.2 x 0.3
Quickbird	Denver, USA	0.6	16.9 x 16.5
IKONOS	Athens, Greece	1	9.7 x 12.3
Orbview 3	Orange, USA	1	0.6 x 0.7

Altogether, five different datasets, coming from airborne and spaceborne sensors, were examined. These datasets depict urban regions with varying building sizes, patterns and densities. It should be mentioned here that only subsets have been used for the investigations.

2. DESCRIPTION OF WORKFLOWS

2.1 Hough Transformation

The proposed workflow for automated building extraction from image data by employing the Hough Transformation has been thoroughly described in Vozikis (2004). Figure 1 shows the major steps of the process.

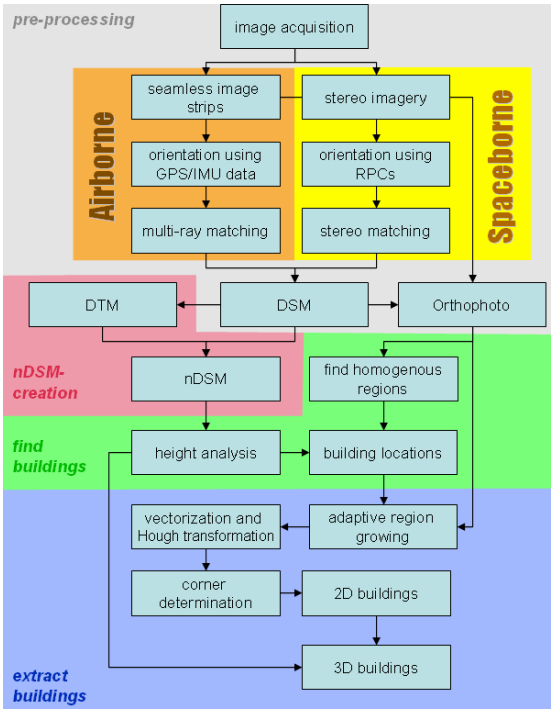


Figure 1: Proposed workflow for automated building extraction from image data

All steps in this workflow are highly automated and human interaction is reduced to a minimum.
In the following the 4 major steps are briefly described.

2.1.1 Pre-Processing

This step comprises the procedures from orientation of the input data up to the DSM (Digital Surface Model) creation. For VHR satellite imagery the orientation approach is based on the RFM (Rational Function Model) (Vozikis et al., 2003). When dealing with aerial imagery it is made use of GPS/INS information in order perform direct georeferencing, and thus automated image triangulation (Scholten and Gwinner, 2003). The DSM extraction is performed by automated correlation procedures, which nowadays are very mature and produce very good results.

2.1.2 nDSM Creation

The goal is to derive the DTM (Digital Terrain Model) from the DSM and subtract it from the DSM in order to produce the so-called nDSM (normalized Digital Surface Model). This way all extruding objects in the data set (including buildings) stand on elevation height 0 (Figure 2).

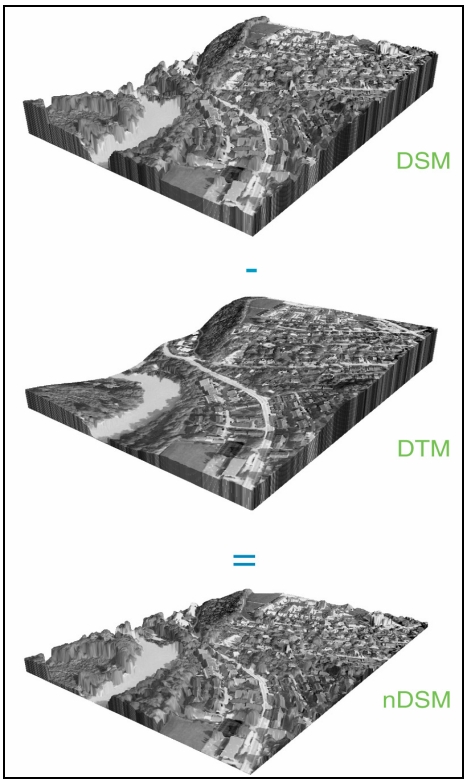


Figure 2: DSM, DTM and nDSM.

2.1.3 Building Detection (Seeding)

This crucial step deals with the identification of potential building candidates in the data sets (=determination of seed points inside buildings). It is proposed to perform 2 statistical analyses. First, perform a thresholding in the nDSM and filter out all objects that are not taller than a certain height, and second, perform texture analysis in the image data to keep only roof-similar regions in the data set (Vozikis, 2004).

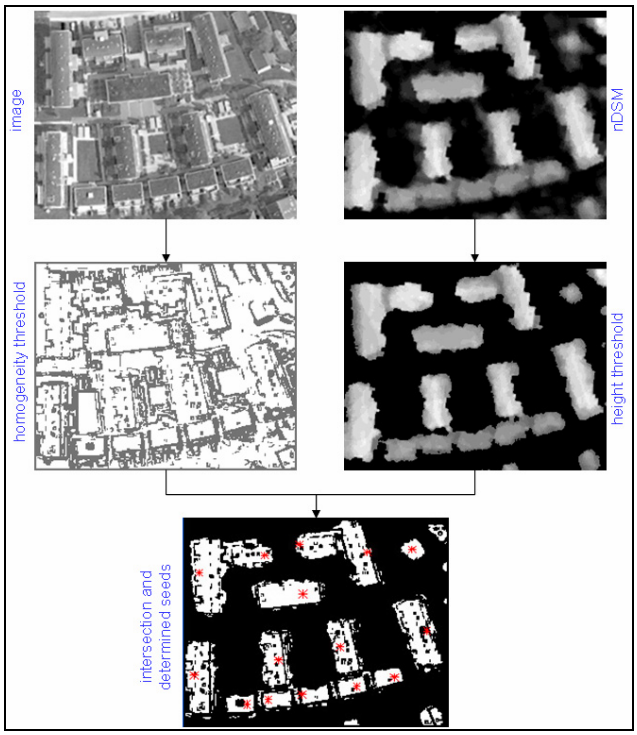


Figure 3: Computation of seed points (red asterisks) inside potential building candidates by height-thresholding and texture filtering.

2.1.4 Building Extraction

By applying the Hough Transformation (to an image of gradient or of contours) the geometric properties of the buildings (building edges and corners) are extracted. Our approach is based on a stepwise, iterative Hough Transformation in combination with an adaptive region growing algorithm (Vozikis and Jansa, 2008). The general idea is to transform the information in the image (feature space) into a parameter space and apply there an analysis. It is a technique for isolating features that share common characteristics. The classical Hough transformation is used to detect lines, circles, ellipses etc., whereas the generalized form can be used to detect features that cannot easily be described in an analytical way.
The mathematical analysis of the Hough Transformation is described in detail in Gonzalez and Woods (1992).
Briefly it can be described as follows:

$$\rho = x\cos(\theta) - y\sin(\theta) \qquad (1)$$

where ρ is the perpendicular distance of a line from the origin and θ the angle (in the range 0 to π) as illustrated in Figure 4.

To apply this function on the whole image, Equation 1 can be extended as shown in Equation 2.

$$H(\theta,\rho) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} F(x,y)\delta(\rho - x\cos(\theta) - y\sin(\theta))dxdy \qquad (2)$$

where δ is the Dirac delta-function. Each point (x,y) in the original image F(x,y) is transformed into a sinusoid ρ = xcos(θ) – ysin(θ).

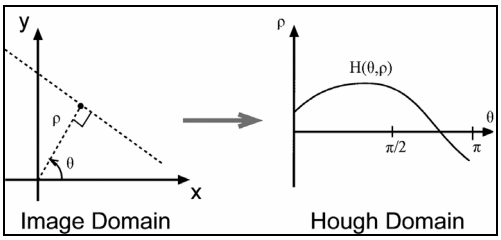


Figure 4: Hough transformation.

Points that lie on the same line in the image (feature space = Image Domain) will produce sinusoids that all intersect at a single point in the Hough domain (parameter space = Hough Domain). For the inverse transform, or back-projection, each intersection point in the Hough domain is transformed into a straight line in the image (Figure 5).

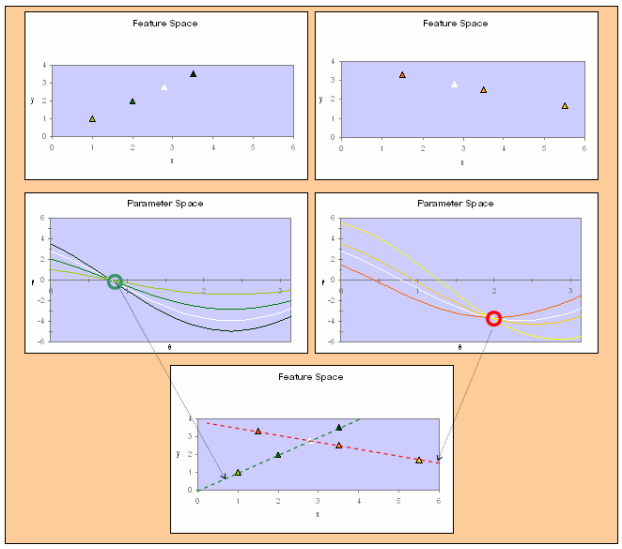


Figure 5: Example: Hough transformation.

The research shows that by using the Hough Transformation for building extraction we have many advantages, such as the good handling of noisy data, the easy adjustment of level of detail of the output data, the ability to force certain geometric properties into the extracted buildings and the possibility to bridge gaps, meaning that building corners that might not be visible in the imagery can be determined accurately. The proposed methodology proves to have certain weaknesses when dealing with radiometrically heterogeneous roofs, when big shadows cover large areas of roofs of the buildings to be extracted, when the building geometry becomes very complex, or when the input data set comprises many compound building (Vozikis and Jansa, 2008).

2.2 Image Matching

This strategy follows the basic principle of image matching by correlation. A given reference image matrix is searched in the image under investigation (the so-called search image) by moving the reference matrix pixel by pixel over the entire image area. Potential candidate positions, i.e. positions of high similarity, are marked if a so-called correlation coefficient exceeds a predefined threshold. In order to find the optimum geometric fit, the searching procedure includes, besides translation, also rotation and scaling. Thus houses of similar shape but different size are found too.

The reference image is usually a small image matrix, here depending on the size of the building to be searched, whereas the search image is a rather big image matrix in our case covering the whole area under investigation.

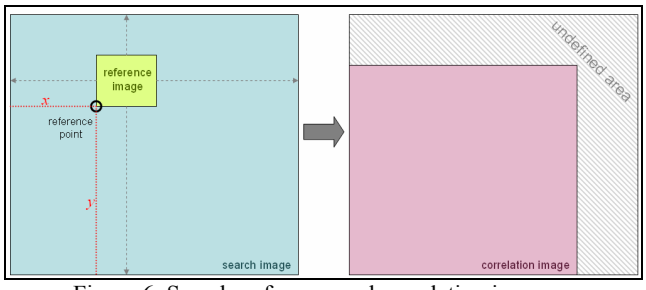


Figure 6: Search, reference and correlation image.

Figure 6 shows the principle of the correlation procedure. The left hand side indicates the searching process with the reference image and the given spaceborne or airborne image as search image. The correlation index is computed for each position of the reference image and the results are stored as similarity measure in the so-called correlation image. Potential building positions are characterized by a high correlation coefficient and thus the correlation image just needs to be thresholded and the local maxima are localised. It has to be mentioned that one crucial parameter is certainly the appropriate threshold value. Its choice determines quite significantly the quality of the result. If the threshold is too low, too many buildings are detected leading to a great number of false matches. If the threshold is too high, the selection is too strict and, as a consequence, too many buildings will be rejected. It is not possible to define an optimum threshold as a general suggestion. For the cross-correlation coefficient using 0.7 to 0.8 is certainly a good choice for starting, but individual adjustments are necessary in any case.

As measure of similarity the cross-correlation coefficient (Equation 3) is adopted, but also other measures can be used (Equations 4 and 5).

$$c_1 = \frac{\sum (g_1 - \bar{g}_1)\cdot(g_2 - \bar{g}_2)}{\sqrt{\sum (g_1 - \bar{g}_1)\cdot\sum (g_2 - \bar{g}_2)}} \quad (3)$$

$$c_2 = \frac{\sum \left(|g_1 - g_2|/(g_1 + g_2)\right)}{n} \quad (4)$$

$$c_3 = \sqrt{\frac{\sum (g_1 - g_2)^2}{n}} \quad (5)$$

where *g1* and *g2* are the grey values in the reference and search window,

$\bar{g}_1$ and $\bar{g}_2$ are the mean grey values in the reference and the search window and

n is the number of used pixels.

Kraus (1996) suggests rewriting Equation 3 as follows for a more efficient computation:

$$c_1 = \frac{\sum g_1 \cdot g_2 - n\cdot\bar{g}_1\cdot\bar{g}_2}{\sqrt{\sum \left(g_1^2 - n\cdot\bar{g}_1^2\right)\cdot\sum \left(g_2^2 - n\cdot\bar{g}_2^2\right)}} \quad (6)$$

Note, that when using Equation 6 the computing effort is reduced since the expression $\sum \left(g_1^2 - n\cdot\bar{g}_1^2\right)$ is constant during the whole process and has to be calculated only once.

The finding of the maxima in the correlation image with sub-pixel accuracy by approximating the discrete correlation function by a continuous polynomial function is broadly discussed in Kraus (1996) and will not be described here.

Figure 7 shows an example were one search image was matched with multiple reference images.

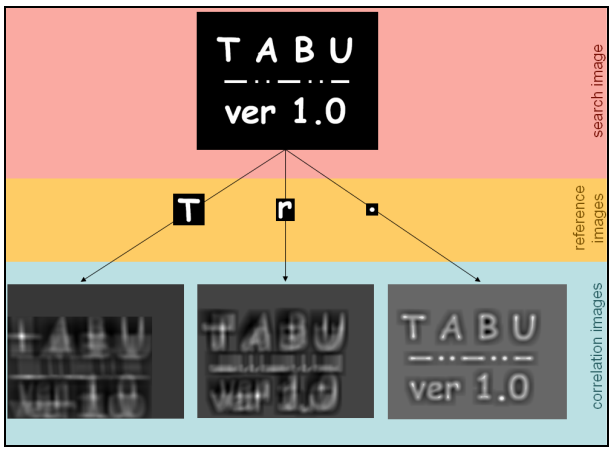


Figure 7: Correlation example. The brighter the pixels, the higher the correlation, i.e. the greater the similarity between search image and reference image.

For the practical implementation, the reference images of buildings are stored in a library. For each of these buildings also vector information (describing the building outline in the reference image coordinate system) is available. Thus the library contains multiple building types to which one image patch and one vector representation corresponds.

Once a location of high correlation is found in the search image, the vector data of this building is transformed into the coordinate system of the search image.

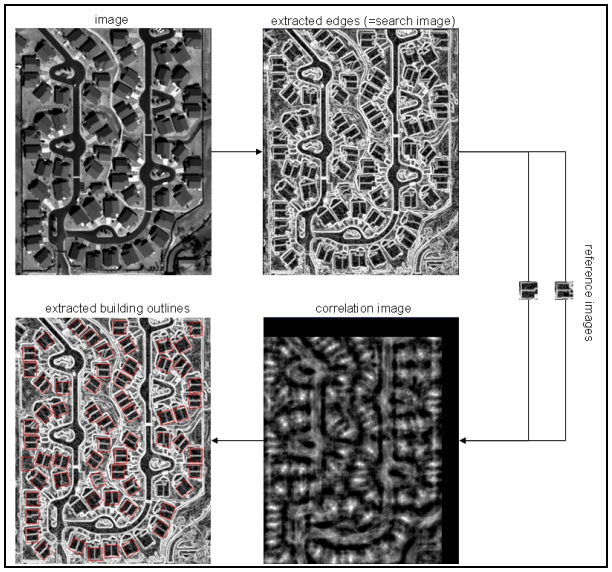


Figure 8: Example on image matching with two reference images and multiple orientations (Quickbird-subset of Phoenix area). Note that the computed correlation image is a multi-dimensional image. The number of dimension corresponds to the number of rotations (here: 120 with a rotation step of 3 degrees).

It is often the case that buildings of the same (or similar) shape have different colours (grey values) in the images (e.g. due to different roof materials). Therefore it is advisable not to store image patches of the investigated buildings in the library, but instead, register their edges. In this case, also the search image has to be edge-extracted before applying the matching procedure. For gaining the edge information, classical operators like the Canny edge detector, Sobel operator, Laplacian of Gaussian etc. can be applied.
Figure 8 shows an example of the image matching procedure.

2.3 Texture Analysis

One of the simplest ways for describing texture is to use statistical moments of grey level histograms of an image or a region. Measures of texture computed using only histograms suffer from the limitation that they carry no information regarding the relative position of the pixels with respect to each other. One way to bring this type of information into texture analysis process is to consider not only the distribution of intensities, but also the distribution of intensity variation (Gonzalez and Woods, 2002).

For this kind of textural examination, firstly the so-called co-occurrence matrix has to be derived for the examined area. This particular matrix holds e.g. information of pixel changes in multiple directions (usually horizontally, vertically and diagonally). The co-occurrence matrix' extents are same in both directions and equal to the number of grey levels that will be considered. For example, for an 8 bit image (256 grey values) the co-occurrence matrix' extents would be 256 by 256. Usually a recoding is carried out to reduce the number of grey value classes (also called bins). A recoding of the original image down to 16 grey levels is for most of the cases satisfying (Gong et al., 1992). Nevertheless, during this research (on texture analysis), all images were recoded to 40 bins.

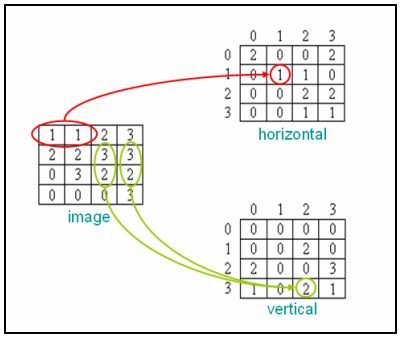


Figure 9: Image and corresponding co-occurrence matrices in horizontal (left to right) and vertical (top to down) directions.

At each position *mc,r* the co-occurrence matrix holds the number of changes from class *r* (=row indices) to class *c* (=column indices) (see Figure 9). This computation is carried out for multiple directions, meaning that one co-occurrence matrix is created for each direction. Figure 10 illustrates the creation of such matrices; here, four grey values exist and the co-occurrence matrices are derived for horizontal and vertical directions.

The task now is to analyze a given co-occurrence matrix in order to categorize the region for which it was computed. Therefore descriptors are needed that characterize these matrices. Some of the most commonly used descriptors are

thoroughly described in (Haralick 1979, Gonzalez and Woods 2002, Zhang 2001). It is obvious how important it is to include some kind of information that tells us whether the values are well distributed over the whole matrix, or whether they are mostly located close to the matrix diagonals (e.g. Difference Moment or Inverse Difference Moment, Equations 5-10 and 5-11). In case big values lie close or on the diagonal (Figure 10, non-urban), the region under investigation is expected to be homogeneous, whereas if the values are distributed more homogeneously (Figure 10, urban), the co-occurrence matrix corresponds to a heterogeneous region.

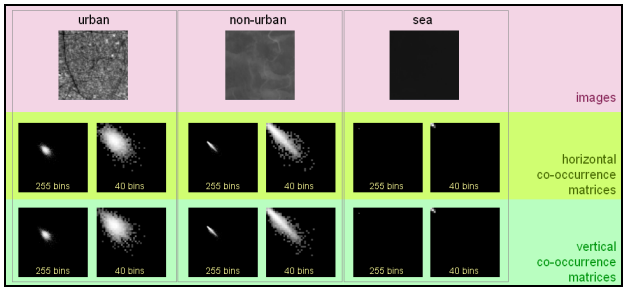


Figure 10: Horizontal and vertical co-occurrence matrices for 3 different types of terrain.

Usually the quality of the input data (i.e. imagery and DSM) is responsible for erroneous results. It is very difficult for the introduced algorithm to produce correct results, if the buildings to be extracted does not cover a certain number of pixels. For instance, when dealing with IKONOS imagery (with GSD of 1m) a small house of 8m x 10m will most probably not be extracted correctly.

The procedures discussed in paragraph 2.2 and 2.3 fall into the category of feature extraction and make use of the imagery for deriving the geometric building properties.
The general idea is to use a library where characteristic features of buildings are stored; by analyzing the image, areas are searched that correspond to a high degree to the registered "library buildings". Characteristic features of a building can be textural measures (by using the so-called occurrence and co-occurrence descriptors) or similarity measures of the image grey values.

3. RESULTS

In this chapter an evaluation of the presented methods is given. It consists of a quantitative and qualitative description, and moreover shortcomings and weaknesses of the presented methods are discussed.

Since many subsets are examined that are coming from various types of line scanning systems, both airborne and spaceborne (ADS40, HRSC-AX, Quickbird, Orbview, IKONOS, SPOT5), their outcomes will not be listed individually. Errors in the qualitative evaluation will be given in image space units (pixels).

We consider that the pre-processing has been carried out without error, so that the orientation of the imagery and the derived orthophotos on which we apply the investigated techniques are correct. We will also not evaluate nDSM extraction algorithms and their qualities in detail, since this is not topic of this research.

The presented outcomes are divided into two groups: quantitative and qualitative results. Moreover, the three presented DCM extraction approaches are evaluated individually. Input data is subdivided into categories depending on image scale and building density (low urban and urban) of the investigated areas. Image scale is defined as the scale that we would expect from an analogue product, e.g. for a 1:10,000 product we expect 1-2 metres accuracy in nature, if the graphical accuracy and visual perceptivity are 0.1-0.2mm.

Regarding the mentioned image scales the three interpretation categories are:

1. scale A: 1:1000-1:4000
2. scale B: 1:4000-1:12000
3. scale C: < 1:12000

3.1 Quantitative Assessment of Building Extraction

The aim in the quantitative analysis is to evaluate whether the presented approaches are practical in sense of completeness of building detection of the result, i.e. how many buildings were actually found. It is investigated whether the techniques for finding potential building candidates are applicable. Furthermore, an evaluation is carried out to see how many of these buildings were extracted and to what a degree:

- **CFB**: **C**orrectly **F**ound **B**uildings,
- **NFB**: **N**ot **F**ound **B**uildings (also includes insufficiently mapped buildings: building seed point was determined successfully, but the adaptive region growing process did not manage to create an area that covers a reasonable amount of the object),
- **WFB**: **W**rongly **F**ound **B**uildings, i.e. found objects were in reality no building exists.

The calculation of CFB (true positive), NFB (false negative) and WFB (false positive) are briefly explained in the following:
The CFB and NFB percentages are calculated with respect to the total number of existing buildings in the area under investigation, whereas the WFB is calculated with respect to the total number of found buildings (comprising correctly and wrongly found buildings). Figure 11 shows the way of computing and a numerical example, respectively.

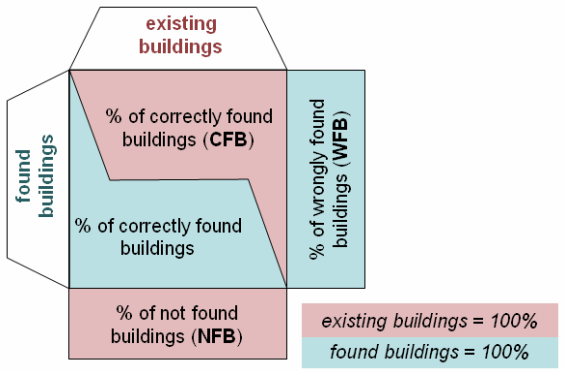


Figure 11: Illustration for quantitative assessment computation.

For the evaluation of the outcomes altogether 13 different scenes containing 677 buildings were examined.
Table 2 allocates the quantitative analysis.

Concerning the level of detail that can be derived from individual data sets the Nyquist theorem has to be taken into consideration ("Sampling rate must be at least twice as high as

the highest frequency of the signal."). For our situation it means that sensor's geometric resolution determines the object resolution, or in other words, the level of detail of the object.

Table2: Quantitative results.

scale	A	B	B	C	C
type	urban	low urban	low urban	low urban	low urban
sensors	airborne	airborne	space-borne	space-borne	space-borne
method	Seeding	Seeding	Seeding	Matching	Texture Analysis
CFB	80.5%	90.2%	97%	88.3%	34.4%
NFB	19.5%	9.8%	3%	11.7%	65.6%
WFB	7.3%	4.8%	30.3%	1.7%	17.2%

3.2 Qualitative Assessment of Building Extraction

The qualitative analysis is based on a comparison between the building outlines derived by using the proposed automated methodology, and manually mapped buildings (image restitution). The manual mapping is carried out by a professional operator who performs 2D (or 3D) digitization on the input data (oriented imagery or orthophotos).

The residuals of each building corner from the manual mapping and the closest point of the automatically extracted shape are computed as a quality measure.

The results are categorized in three groups depending on the method used for building extraction (Table 3). The RMS is given in pixels.

Table3: Qualitative results.

	Hough	Matching	Texture Analysis
RMS x	0.937	0.898	0.954
RMS y	0.914	0.958	0.996
total RMS	1.309	1.313	1.379

The number of examined objects is the same as in the quantitative analysis.

Note that the figures in Table 3 are based on image residuals. They show the difference of the automatically derived corner points and the digitized ones in the image. As our data sets were acquired with vertical viewing angles these results can be also interpreted as planimetric object space residuals.

But when dealing with images that were captured with oblique viewing angles, the buildings must be projected into object space in order to carry out a qualitative analysis in the reference system.

4. CONCLUSIONS

The aim of this work was to propose a method for generating DCMs which makes use of images from spaceborne or airborne line scanning devices, on orthophotos if available and on elevation models. Various image processing techniques, such as Hough transformation, adaptive region growing, image matching, texture analysis, were employed and investigated for deriving the strengths and weaknesses of each. A variety of data sets were tested, coming from both spaceborne and airborne acquisition systems. Through the research based on adaptive region growing and on the iterative Hough transformation we can conclude that the method is very powerful, but has also some weaknesses. One is the high dependence on the radiometric quality of the input imagery. Furthermore, rather small buildings will not be treated correctly. Image matching proved to be a very effective, but very time consuming. The suggested strategy of texture analysis, although very efficient for pattern recognition over areas in small scale imagery, was not very successful for extracting individual buildings.

Through this research partly very good results were obtained, but nevertheless further investigations are necessary for improving the quality of the results even more.

Future work will be focused on:

- Extraction of objects with holes (e.g. houses with inner courtyards), i.e. deriving the inner and outer boundary of buildings.
- Research on constraint settings for aggregating neighbouring roof parts that belong to one building.
- Introduction of multispectral information for making the algorithms more efficient, especially as far as seed point determination is concerned.
- Extract edges on sub-pixel bases.
- Integrate a hierarchical approach in order to decrease computation time.

5. REFERENCES

Gong, P., Marceau, D.J. and Howarth, P.J., 1992. A Comparison of Spatial Feature Extraction Algorithms for Land-Use Classification with SPOT HRV Data. Remote Sensing of Environment, Vol. 40, pp. 137-151.

Gonzalez, R.C., and Woods, R.E., 1992. *Digital Image Processing*. Reading, MA: Addison Wesley.

Gonzalez, R.C. and Woods, R.E., 2002. Digital Image Processing, Second Edition. Patience Hall, New Jersey, 799 pages.

Haralick, R.M., 1979. Statistical and Structural Approaches to Texture. Proceedings IEEE, Vol. 67, No.5, pp. 786-804.

Kraus, K., 1996. Photogrammetrie, Band2, Verfeinerte Methoden und Anwendungen. Duemmler Verlag, Bonn, 488 pages.

Scholten, F. and Gwinner, K., 2003. Band 12 "Publikationen der Deutschen Gesellschaft für Photogrammetrie, Fernerkundung und Geoinformation", Bochum, pp. 419-426.

Vozikis, G., Jansa, J. and Fraser, C., 2003. Alternative Sensor Orientation Models for High-Resolution Satellite Imagery. *Band 12 "Publikationen der Deutschen Gesellschaft für Photogrammetrie, Fernerkundung und Geoinformation",* Bochum , pp. 179- 186.

Vozikis, G., 2004: Automatic Generation and Updating of Digital City Models using High-Resolution Line-Scanning Systems. The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Vol. 34, Part XXX.

Vozikis, G. and Jansa, J., 2008. Advantages and Disadvantages of the Hough Transformation in the frame of Automated Building Extraction. International Society for Photogrammetry and Remote Sensing XXIst Congress, Beijing, China; 07-03-2008 - 07-11-2008; in: "Proceedings", Vol. XXXVII. Part B3b (2008), ISSN: 1682-1750; 719 - 724.

Zhang, Y. (2001): A Spectral and Spatial Information Integrated Approach for Tree Extraction from High-Resolution Digital Imagery. Digital Earth, Fredericton, Canada, 9 pages.

SEMI-AUTOMATIC CITY MODEL EXTRACTION FROM TRI-STEREOSCOPIC VHR SATELLITE IMAGERY

F. Tack [a,*], R. Goossens [a], G. Buyuksalih [b]

[a] Dept. of Geography, Ghent University, Krijgslaan 281, 9000 Ghent, Belgium – (f.tack, rudi.goossens)@ugent.be
[b] IMP-Bimtas, 34430 Beyoglu, Istanbul, Turkey – gbuyuksalih@yahoo.com

KEY WORDS: Photogrammetry, City DSM generation, Tri-stereoscopy, Ikonos, Urban

ABSTRACT:

In this paper a methodology and results of semi-automatic city DSM extraction from an Ikonos triplet, is introduced. Built-up areas are known as being complex for photogrammetric purposes, mainly because of the steep changes in elevation caused by buildings and urban features. To make surface model extraction more robust and to cope with the specific problems of height displacement, concealed areas and shadow, a multi-image based approach is followed. For the VHR tri-stereoscopic study an area extending from the centre of Istanbul to the urban fringe is chosen. Research concentrates on the development of methods to optimize the extraction of a surface model from the bundled Ikonos triplet over an urban area, without manual plotting of buildings. Optimal methods need to be found to improve the radiometry and geometric alignment of the multi-temporal imagery, to optimize the semi-automatical derivation of DSMs from an urban environment and to enhance the quality of the resulting surface model and especially to reduce smoothing effects by applying spatial filters.

1. INTRODUCTION

The high level of detail and geometric accuracy of very high resolution satellite data such as Ikonos imagery, has made this kind of imagery suitable for DSM generation at feature level of urban environments. Due to the photogrammetric complexity of urban areas, quite some research is done to cope with the specific problems of urban surface model generation from standard stereopairs. As a multi-image based approach can make the 3D modelling more robust, a methodology and results of semi-automatic DSM production from an Ikonos triplet over an urban area, is highlighted in this paper. From a theoretical point of view the redundancy of a third image should lead to a more reliable photogrammetric processing. Only a few investigations have been published dealing with the concerning subject. Research published in (Baltsavias et al., 2006) and (Raggam, 2006) can be referred to.

Research is conducted within the framework of the MAMUD project (Measuring And Modelling of Urban Dynamics) funded by the STEREO (Support to The Exploitation and Research of Earth Observation data) program of Belgian Science Policy. The objectives of the MAMUD research project is to investigate the possibilities of earth observation for a better monitoring, modelling and understanding of urban growth and land-use change. Urban change processes are affecting the human and natural environment in a not unimportant way. This enlarges the need for more effective urban management approaches based on sustainable development. A sustainable urban management needs sufficiently detailed and reliable base information on the urban environment and its dynamics. Satellite imagery has proven to be an important data source to monitor and describe urban areas and its changes. Hereby, detailed information on the vertical structure is vital to label urban features, to describe urban morphology and to generate spatial metrics. If the subsequent approach is proved to be successful, it will increase the flexibility of producing semi-automatic 3D city models from high resolution satellite imagery.

The complexity of an urban environment for photogrammetric purposes will be highlighted in section 2. In section 3 the image dataset and work area will be outlined. The different phases of the photogrammetric processing of the Ikonos triplet are stated in section 4. In following section, spatial filtering is applied on the height values of the surface model to improve the quality and reduce smoothing effects. Geometric accuracy analysis is discussed in section 6. Finally, in section 7 experiences and conclusions are summarized.

2. COMPLEXITY OF URBAN AREAS

A Digital Surface Model is a digital representation of the terrain and topographic object height in a grid structure. Interpolation of the discrete height values is needed to approximate the continuity of the ground surface. Urban environments are experienced as complex for 3D modelling purposes because of the steep changes in elevation and the discrepancy between the smoothness of the ground surface and abrupt discontinuities caused by buildings and other urban features. Without manual plotting or spatial filter techniques it is difficult to reconstruct vertical walls out of VHR satellite imagery. An interpolation technique creates a smoothed surface and causes individual buildings will have a shape of a bell instead of the rectangular geometry (Jacobsen, 2006). A second consequence of steep changes in elevation is the occurrence of shadow and concealed areas. Due to the convergent viewing angle of VHR sensors like Ikonos, terrain features with certain height above the surface are geometrically displaced in the imagery, leading to dissimilarities between the stereo images.

By this distortion of its true position, parts of the ground surface can be hidden in the satellite image. These are so-called occluded areas. Shadow areas, which have poor contrast, and

* Corresponding author

occlusion areas lead to mismatches during the image matching algorithm and subsequently to errors in the resulting surface model.

3. DATA SET AND STUDY AREA

The satellite Ikonos is able to rotate the CCD Linear Array sensor up to an angle of 26° off-nadir, so the satellite can take images of the same location from two different view points on the same orbital track. Next to along track stereo pairs, it is also possible to create stereo couples out of images from the same area but taken from a different orbit at a different date. These are so-called across track stereo pairs. This approach to form couples has some disadvantages. The most important ones are radiometric differences and changes of the ground surface due to the time gap between acquisition of the imagery. A triplet is constructed out of an along track Ikonos stereo pair taken in March 2002 and a third image taken in May 2005. The third image can be considered as a nadir image. Selection criteria for the near vertical image were multiple: overlap with stereo couple, cloud-free acquisition, minimal time interval and optimal stereo constellation. Despite the big time interval, the 2005 Ikonos image was chosen to be the most optimal candidate.

Figure 1. High resolution study field, indicated by the red polygon. The extent of the HR study area covers the overlapping area between the 3 Ikonos images.

The Ikonos STEREO product imagery, which comprises of a forward and backward image acquisition and the GEO Ortho Kit 2005 image are panchromatic, resampled to a spatial resolution of 1 m by the image provider and provided with the Rational Polynomial Coefficients (RPC) camera model file. Further characteristics of each image of the triplet can be found in table 1.

Image ID	Acquisition date	Elevation angle	Collection azimuth	Sun elevation angle
A (Forward)	1/03/2002	67.59°	1.6°	39.1°
B (Backward)	1/03/2002	75.59°	214.1°	39.1°
C (Nadir)	16/05/2005	80.93°	23.5°	65.5°

Table 1. Characteristics of the three VHR satellite images acquired over the study field.

Part of the mega city Istanbul, Turkey is chosen as test field for the project, mainly because it is a city characterized by an intense urban growth. The city is very compact and concentrated along the Bosphorus strait. The high resolution test area covers the overlapping area between the Ikonos 2002 stereo pair and the 2005 image and covers an area of approximately 60 km^2, containing Istanbul's historic peninsula and going up to the north to the urban fringe. It concerns a densely built-up area with a height range of 220 m with the lowest point at sea level and geo-morphologically characterized by a hilly landscape.

4. SURFACE MODEL GENERATION

In following subsections, the successive steps of the applied methodology for city surface model generation, based on (tri)stereoscopic VHR satellite imagery, are elucidated. The emphasis is especially laid on those phases were research is done to cope with the complexity of an urban environment.

4.1 Tri-stereoscopic approach

Instead of the standard stereo mapping with two images a tri-stereoscopic approach is followed. Generation of a DSM using more than two overlapping images has some interesting characteristics. First of all, this approach strengthens the image orientation because of the redundancy in the geometric reconstruction. Points in object space can be calculated by the best fit of N convergent image rays instead of two. Secondly the redundancy leads to a more robust matching, as mismatches and a unique solution, in case of multiple matching candidates, can be easier identified. In the stereo case, an object point cannot be matched if it is located in an occluded area on one or both images. In the tri-stereoscopic case, the third image is taken from a different viewing angle. Consequently this leads to a shift of the occluded areas in the image and enlarges the chance of a successful match.

Processing of the Ikonos triplet is mainly done with a photogrammetric software platform, called SAT-PP. SAT-PP is able to perform image matching on more than two images simultaneously (Zhang & Gruen, 2006). This is in contrast to most photogrammetric software packages that are only able to match two images at the same time.

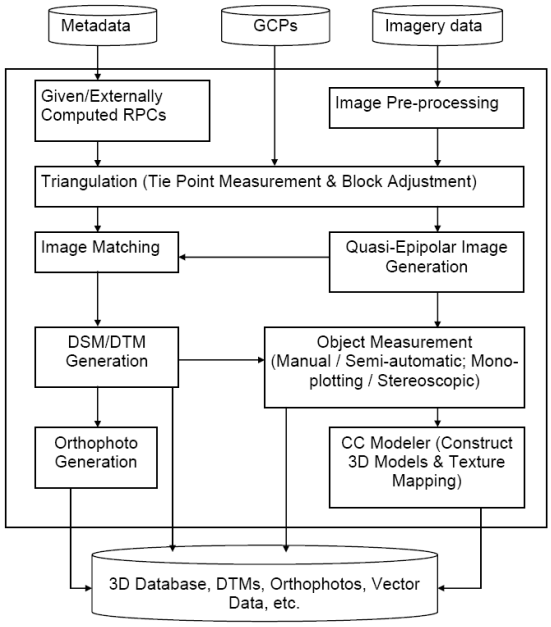


Figure 2. SAT-PP workflow © Chair of Photogrammetry and Remote Sensing ETH Zurich.

4.2 Preprocessing of the satellite data

Before processing the VHR imagery a contrast enhancement is executed as this leads to a more reliable image matching. Especially between images of the same area but taken at different dates from different orbits large radiometric dissimilarities can occur due to different illumination and atmospheric conditions, leading to poor matching results. To enhance the contrast for each image individually and to equalize the radiometric differences between the imagery, a Wallis filter was applied (Wallis, 1976).

The general form of a Wallis filter is given by:

$$g^{w}(x,y) = g(x,y) * r_1 + r_0 \quad (1)$$

$$r_1 = \frac{cs_h}{cs_g + (s_h / c)} \quad (2)$$

$$r_0 = bm_h + (1 - b - r_1)m_g \quad (3)$$

with $g^w(x,y)$ and $g(x,y)$ = filtered and original image
m_g and s_g = original mean and standard deviation values
m_h and s_h = target value for mean and standard deviation
c and b = contrast expansion and brightness forcing c^{te}

The Wallis filter performs a non linear, locally adaptive contrast enhancement. Actually a large kernel divides the image in different sections and within each section the local contrast is optimized. Applying a Wallis filter on the original images does not only result in an enhancement and sharpening of texture patterns in areas of low contrast and equal overall contrast but normalizes also the radiometry, especially between images taken at different dates. The effect of radiometric enhancement of very high resolution satellite imagery is illustrated in figure 3 & 4. The Wallis filter enhances existing texture patterns, leading to optimization of the contrast in shadow areas. Note that in the shadow rich areas axis-aligned artefacts are introduced due to the Wallis filtering.

Figure 3. Extract of original 11-bit Ikonos image, illustrating an area with high buildings. There is very little contrast within the shadow areas, leading to mismatches during the image matching process.

Figure 4. Extract of Wallis-filtered 11-bit Ikonos image. The radiometric filter enhances the existing texture patterns locally, leading to optimization of the contrast in the shadow areas.

Also an adaptive smoothing filter is applied to reduce image noise while sharpening edges. As noise is an important data-source for mismatches, reducing it further improves the quality of the surface model.

Next to the radiometric enhancement a method for geometric normalization was devised. The Ikonos 2002 stereo couple is epipolar projected and suitable for stereo applications. As the 2005 Ikonos image is taken from a different orbit, the images are displaced to each other and the internal geometry will be slightly different because of the different scan direction. Geometric normalization of the 2005 Ikonos image with the

2002 STEREO product imagery is done by image co-registration in ENVI. The 2005 image is resampled according a first-order polynomial transformation to geometrically align the multi-temporal imagery. A first-order polynomial transformation corrects for rotation, translation, scaling and shearing. As the orientation of the 2005 image has changed after registration, it was necessary to calculate a posteriori RPCs for the resampled image, which is not a straightforward task. Ad hoc RPC generation was done in collaboration with a team of Prof. Dr. Crespi from the Area di Geodesia e Geomatica, La Sapienza University of Rome. An algorithm, developed and embedded in the software package SISAR (Software per Immagini Satellitari ad Alta Risoluzione), makes it possible to generate RPCs starting from physical sensor models, image metadata, transformation parameters and a set of 15 to 20 ground control points with known map coordinates (Bianconi, 2008 and Crespi, 2009). Image coordinates for the GCPs were collected on the original and resampled 2005 Ikonos image. Based on this method, RPCs could be generated with an accuracy of 3.8 pixels in line direction and 5.1 pixels in sample direction.

4.3 Bundle adjustment for image orientation

During the bundle adjustment process, the rotation along the three axes and position of the sensor during image capturing is calculated for all images simultaneously according a least-squares matching. At the same time the relationship between image and object space is described. To calculate the best fit for all images, initial values for internal and external orientation are needed though. As no information on the physical camera model of Ikonos is released, rational polynomial coefficients, provided by the image vendor, are used to calculate initial values for internal and external image orientation. The rational polynomial function model uses a general polynomial transformation to describe the mathematical relationship between object and image space, instead of a physical sensor model. The rational function model is the ratio of two polynomials and is derived from the physical sensor model and on-board sensor orientation (Grodecki & Dial, 2003).

As RPCs are calculated from on-board sensor orientation data, satellite ephemeris and star tracker observations, the accuracy of image orientation can be refined by using ground control points. During a field trip to Istanbul the necessary GCPs for photogrammetric processing of the DSM's were collected in close collaboration with the Istanbul Metropolitan Planning Centre (IMP-Bimtas). Because accurate large-scale ortho-images were available for the study area and because of the difficulties of GPS measurements in the narrow streets of the densely built-up area, an approach was chosen to derive the GCP from ortho-images supplemented with 1:5000 scale topographic maps. 37 clearly visible GCPs were derived, homogeneous distributed over the study area. In total, 17 points with known map coordinates and clearly identifiable in all three images were used to describe the relationship between the imagery and terrain. The a priori geometric accuracy for the DSM extraction consists of an overall RMSE value of 0.79 m for X residuals, 0.78 m for Y residuals and 2.36 m for Z residuals.

4.4 Epipolar geometry

Before extracting the surface model, the original images are resampled to an epipolar orientation. Y-parallax is removed, while leaving the parallax in X-direction unresolved, which can be interpreted as height differences. This reduces the process of finding conjugate points in overlapping images from a two-dimensional to a one-dimensional search algorithm along epipolar lines.

4.5 Multi-image matching

During the image matching process conjugate features need to be found automatically between the overlapping images. The surface model can be processed afterwards by calculation of height differences based on the measurement of the disparity between corresponding pixels. The applied algorithm works according a coarse-to-fine hierarchical matching strategy. Image pyramids consist of different versions of an image at exponentially decreasing resolutions. The bottom level of the pyramid contains the original image. The matching results of each higher pyramid level are used as approximations in the successive, lower level. At each level also an intermediate DSM is generated from the matched features and is refined through the image pyramid. Based on all data in each pyramid level, the matching parameters are fine-tuned progressively.

The matching algorithm is a combination of feature point, grid point and 3D edge matching. This redundancy leads to better constraints and more reliable results. Grid point matching is especially valuable in areas with less texture where conjugate feature points are hard to detect. For each grid point to be matched in the first image, the matching algorithm searches for the conjugate pixel in the other images that correlates the most by shifting a kernel of certain size along the epipolar line. A correlation constraint is used to identify possible matching candidates. The geometrically constrained cross-correlation or GC^3 method is an extension of the standard cross-correlation technique (Zhang & Gruen, 2006). In case of more than one matching candidate, the information of multiple images, i.e. more than two, can provide geometric constraints which assist to identify a unique matching solution.

3D edge matching is extremely valuable when dealing with urban areas, as they assist in modelling surface discontinuities. Edges are detected by the Canny operator (Canny, 1986). During surface model generation the matched edges will be taken into account as break lines to avoid smoothing effects. In Figure 5, illustrating matched edges in an urban area on Ikonos imagery can be seen that the main shape of most of the buildings is estimated quite well by detected edges. An important source of errors in edge detection is caused by building shadows. As shadow areas are being into large contrast with the surrounding pixels, edges will be detected at the shadow borders.

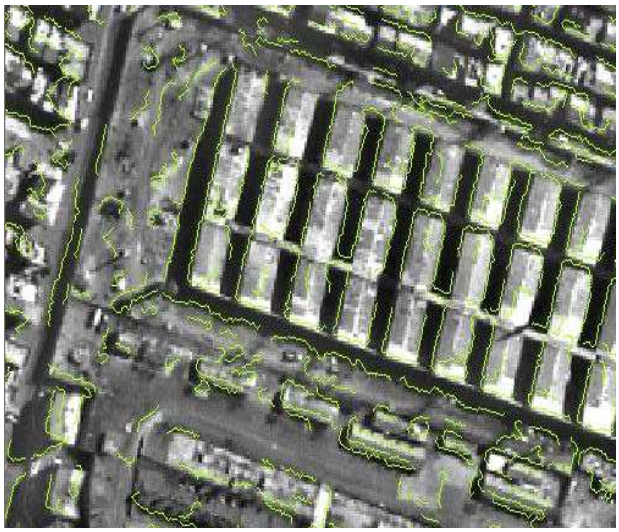


Figure 5. Edge detection & matching in an urban environment on Ikonos imagery.

At a final stage a least-squares matching method, called modified multi-photo geometrically constrained matching algorithm, is performed using all matched points as approximations to detect mismatches and to further refine matching results. The MPGC algorithm combines the matched points with geometrical constraints, derived from multi-image ray intersection conditions and knowledge about the image orientation (Baltsavias, 1991). A Least Squares B-Spline Snakes is used to refine the matched edges. For more details on the matching strategy we can refer to (Zhang & Gruen, 2006).

During image matching, calculation of the position and height of each point or line is treated independently. To create a connected surface, the discrete measurements are interpolated. The resulting surface model is processed at a grid size of 3 meters. The chosen resolution leads to the best equilibrium between detail and reduction of noise. As illustrated in figure 6 & 7, the shape of big buildings and free-standing buildings is modelled well, while in the very dense urban area small buildings are merged into building blocks.

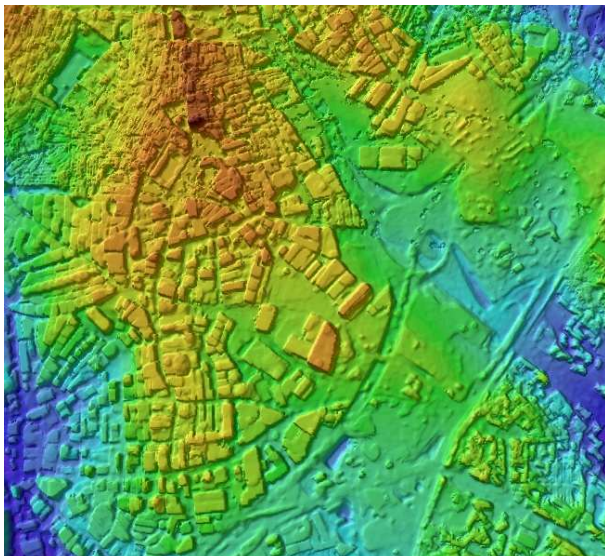


Figure 6. Map view on extract of the 3m colour-coded DSM.

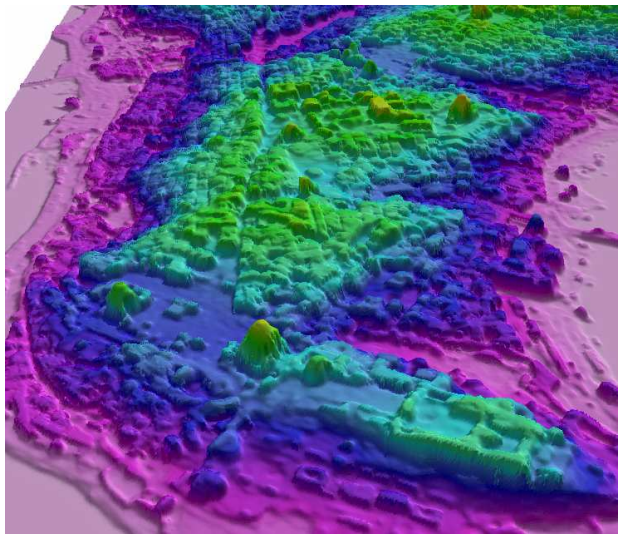


Figure 7. Perspective view on extract of 3m color-coded DSM. The surface model represents Istanbul's historic peninsula.

4.6 Ortho-generation

During ortho-generation phase the sensor geometry of the images, characterized by a parallel projection in along-track direction and perspective projection in across-track direction, can be transformed to map geometry based on the developed surface model. The surface model represents each pixel in its correct geometric position. Back-projection from the DSM to the image supplies the grey value or texture for the pixel. In case of an occluded pixel on the master image, texture information is extracted from a slave image or neighbourhood pixels in case of occlusion on all images. A ground sample distance of 1 m or 1 pixel is chosen for the ortho-image.

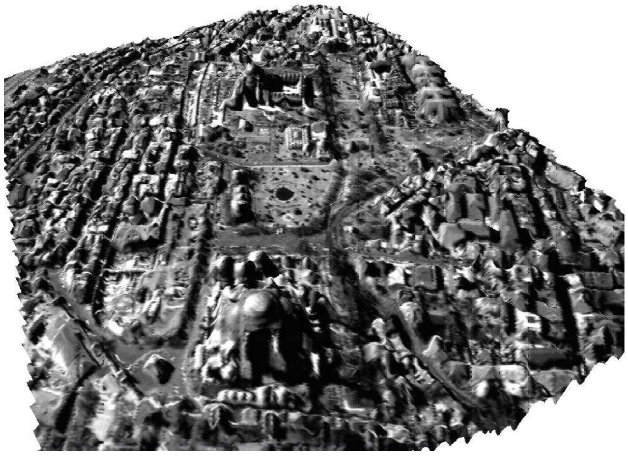


Figure 8. Extract of 3m surface model, draped with panchromatic ortho-image for photorealistic visualization. The surface model represents Istanbul's historic peninsula.

Figure 9. DSM draped with ortho-image, representing an industrial area at the urban fringe.

5. SPATIAL FILTERING

To further improve the global quality of the surface model and especially to reduce smoothing effects, spatial filtering is applied on the height values of the DSM. In a first approach, an order statistics filter is applied on the surface model. More specific a small 7 by 7 median filter is used, which not only reduces noise and outliers but also enhances edges. The value of each pixel is changed by looking at the surrounding pixels within the 7 by 7 kernel and arranging all values in sequential order. Next, the 50th percentile value is assigned to the centre pixel. As the median value is assigned, the influence of outliers within the moving window will be reduced. The outcome of applying a median filter on an urban surface model is further discussed in (Jacobsen, 2006).

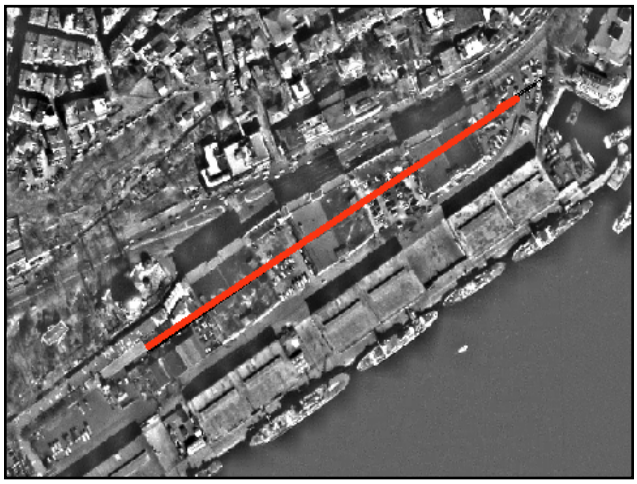


Figure 10. Position and orientation of profile A through 3 similar buildings.

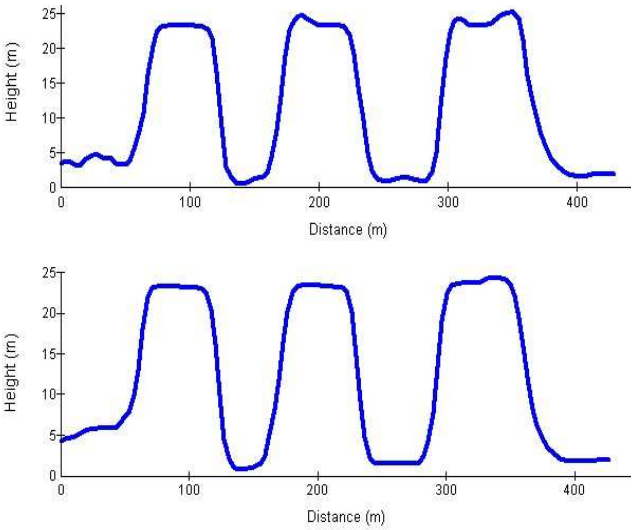


Figure 11. Graphs illustrating profile A before and after median filtering of the 3 m resolution tri-sterescopic surface model. After median filtering, local variations and outliers are reduced and the rooftops are at a more or less constant level.

A method is also developed to further improve building shapes based on the knowledge of building contours. Flat roofs can be assumed for the buildings within the study field. A first attempt failed, where the matched edges were used as approximations for building contours. As can be derived from figure 5, the extracted edges are not closed polygons and sometimes they are connected together with edges of neighbouring buildings. This made the conversion to individual building contours extremely complex.

The results of a second approach are more effective. An external dataset is used, consisting of 2D building footprints which were plotted on aerial imagery by IMP-Bimtas for cadastral purposes. Fitting of the 2D building footprints on the generated surface models, allows to extract all man-made objects. Subtraction of the DSM with the generated building model results in a terrain model (DTM) with gaps where the buildings were positioned. Distinction between a terrain model layer and a building model layer allows to apply different spatial filters adapted to the specific needs of the layer. The terrain model without man-made objects should be a continuous and smooth surface. As smoothing constraints are very important for the DTM, a median filter with a large kernel size of 18 by 18 pixels is used. On the other hand, smoothing must be minimized for the building layer to model shape and discontinuities of man-made objects as good as possible. As the “bell-formed” shape of buildings in an unfiltered surface model is mainly an underestimation of height, an upper quartile filter with a small kernel of 7 by 7 is applied on the building layer two times within the boundaries of each footprint. An upper quartile filter is a nonlinear, order statistics filter and returns the 75th percentile value within the kernel. Spatial filtering of the height values within each building footprint reduces the local variations and puts the roof height on a more or less constant level. In a final step the DTM is merged with the building layer to obtain a final filtered DSM.

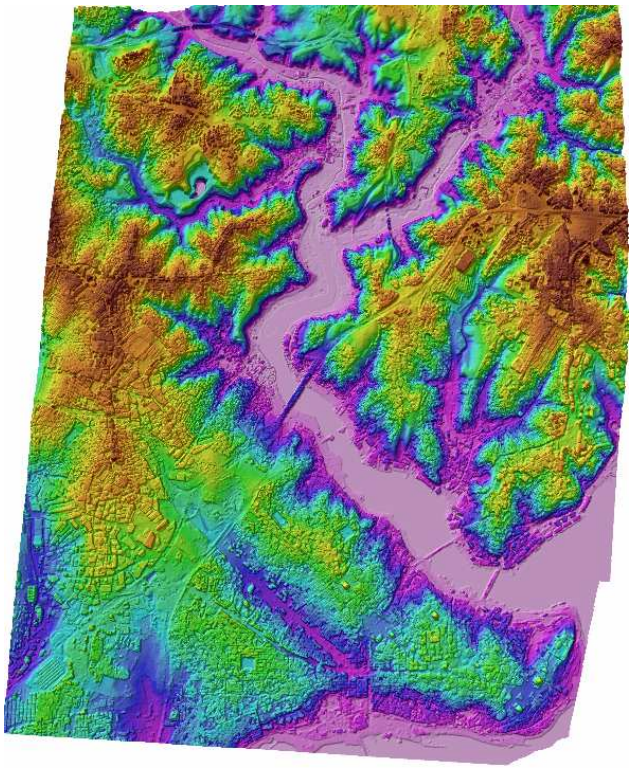


Figure 12. Combination of DTM and building layer to a final DSM, covering the high resolution study field.

6. GEOMETRIC ACCURACY ANALYSIS

A dataset of 35 check points, measured with GPS and evenly distributed over the study area, is used to check the geometric accuracy and quality of the extracted models from stereopair and triplet. It concerns independent ground control points, meaning that they are not used in the photogrammetric processing of the models. Check points are preferred because of the lack of a more accurate reference surface model. Besides, the uncertainty of height errors in a reference map is much bigger than for discrete measured values in the field. Comparison of a measured height value and the calculated height value in the model at a certain location gives statistical information about the accuracy by which reality is modelled.
Some calculated statistics, quantifying the geometric accuracy are presented in table 2. Distinction is made between the standard stereoscopic and tri-stereoscopic approach. The a priori geometric accuracy reflects the quality and robustness of the image orientation. RMS error in X, Y and Z is given for the total of 17 ground control points that were used to fix the mathematical relationship between image and object coordinate space. For X and Y, sub-pixel accuracy is obtained in both approaches. RMSE for the Z component is less than 3 pixels. 35 independent check points are used to calculate the RMS error for Z and the mean Z difference between measured and calculated value by the model. For both statistics the value is less than 3 pixels.

	A priori geometric accuracy				DSM geometric accuracy		
Imagery	No. of GCP	RMSX (m)	RMSY (m)	RMSZ (m)	No. Of CP	RMSZ (m)	Mean dZ (m)
Stereoscopic	17	0.68	0.72	2.44	35	2.61	2.21
tri-stereoscopic	17	0.79	0.78	2.36	35	2.47	2.06

Table 2. Geometric accuracy analysis.

Visual analysis of the models shows big improvements of the quality for the surface model derived from the Ikonos triplet. Noise is reduced and smoothing effects of man-made object are reduced to a minimum, however the improvements do not reflect in the quantitative accuracy check. The RMSE and mean values are slightly better for the triplet than for the stereopair. This is due to the fact that the improvements are mainly situated around buildings and other steep changes in elevation. Check points are mostly measured in open terrain so that they are clearly identifiable on the imagery. Within these non-complex areas the surface model from the stereopair gives also optimal results. To have a better quantification of the improvements, future work should involve the collection of rooftop heights for a set of buildings and comparison between the collected ground truth and the produced models.

7. CONCLUSION

In this treatise an approach is proposed to extract an urban surface model in a semi-automatic way directly from multi-scopic Ikonos imagery, in contrast to surface models derived from manual plotting of building rooftops. The input of the operator during photogrammetric processing is reduced to a minimum. Interesting advantages are that it is less labor-intensive and that the outcome is independent from human interpretation. Off course manual plotting of buildings will lead to a higher accuracy and more detailed information, but this task is very time consuming and will not be cost-effective in some situations. As from the perspective of the geometric accuracy, as from the visual analysis we can conclude that the outcome is encouraging and that acceptable results are reached. At different levels of the photogrammetric processing of the imagery, efforts are done to cope with the complexity of modeling an urban environment. Occlusion and consequently mismatches are reduced by combining the redundant information of a third image with a stereopair. Radiometric and geometric dissimilarities between the multi-temporal imagery are diminished by preprocessing the individual images. Combination of three different matching algorithms gives redundancy and geometric constraints leading to dense and reliable matching results. Finally, spatial filtering is applied on the height values of the DSM to reduce smoothing effects and enhance global DSM quality.

REFERENCES

Baltsavias E., 1991. Multiphoto geometrically constrained matching. PhD Dissertation, Report No.49, Institute of Geodesy and Photogrammetry, ETH Zurich, Switzerland.

Baltsavias E., Pateraki M., Zhang L., 2001. Radiometric and geometric evaluation of Ikonos GEO images and their use for 3D building modelling. *Proc. Joint ISPRS Workshop High Resolution Mapping from Space 2001*, Hannover, 19-21 September, 2001.

Baltsavias E., Zhang L., Eisenbeiss H., 2006. DSM generation and interior orientation determination of ikonos images using a testfield in Switzerland. *Photogrammetrie, Fernerkundung, Geoinformation*, (1), pp. 41-54.

Bethel J.S., McGlone J.Ch., Mikhail E.M., 2001. *Introduction to Modern Photogrammetry*, John Wiley & Sons, Inc., New York, 477 p.

Bianconi M., Crespi M., Fratarcangelli F., Giannone F., Pieralice F., 2008. A new strategy for rational polynomial coefficients generation. *Proceedings of the EARSeL Joint Workshop "Remote Sensing – New Challenges of High Resolution"* Bochum (Germany), March 5-7, 2008, pp. 21-28.

Buyuksalih G., Jacobsen K., 2007. Digital surface models in build up areas based on very high resolution space images. *ASPRS 2007 Annual Conference*, Tampa, Florida, 7-11 May, 2007.

Canny J. F., 1986. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 8 (6), pp. 679-698.

Crespi M., Fratarcangeli F., Giannone F., Pieralice F., 2009. Chapter 4 - Overview on models for high resolution satellites imagery orientation. In: Li D., Shan J., Gong J. (Eds.), *Geospatial Technology for Earth Observation data*, Springer, Heidelberg.

Devriendt D., Goossens R., Dewulf A., Binard M., 2003. Improving spatial information extraction for local and regional authorities using Very-High-Resolution data – geometric aspects. *High Resolution Mapping from Space 2003*.

Grodecki J., Dial G., 2003. Block Adjustment of High-Resolution Satellite Images Described by Rational Functions. *Photogrammetric Engineering and Remote Sensing* 69(1), pp. 59-70.

Jacobsen K., 2005. Analysis of Digital Elevation Models based on space information. *New strategies for European Remote Sensing*, Rotterdam : Millpress, pp. 439-451.

Jacobsen K., 2006. Digital surface models of city areas by very high resolution space imagery. *Workshop of the SIG Urban Remote Sensing*, Berlin, Germany, 2-3 March, 2006.

Krauss T., Reinartz P., Lehner M., Schroeder M., Stilla U., 2005. DEM generation from very high resolution stereo data in urban areas. *5th International Symposium Remote Sensing of Urban Areas*, Tempe, AZ, USA, 14 – 16 March 2005.

Raggam H., 2006. Surface mapping using image triplets : Case studies and benefit assessment in comparison to stereo image processing. *Photogrammetric engineering and remote sensing*, vol. 72, n° 5, pp. 551 – 563.

Taillieu K., Goossens R., Devriendt D., De Wulf A., Van Coillie S., Willems T., 2004. Generation of DEMs and orthoimages based on non-stereoscopical IKONOS images. *Proceedings of the 24th EARSeL symposium*, Dubrovnik, Croatia, 25 – 27 May 2004, pp. 453 – 460.

Wallis R., 1976. An approach to the space variant restoration and enhancement of images. *Proceedings of Symposium on Current Mathematical Problems in Image Science*, Naval Postgraduate School, Monterey, CA, 641-662.

Zhang C., Fraser C., 2008. Generation of digital surface model from high resolution satellite imagery. *Proceedings of XXIth ISPRS congress*, Beijing, China, 3 – 11 July 2008, pp.785-790.

Zhang L., Gruen A., 2006. Multi-image matching for DSM generation from Ikonos imagery. *ISPRS Journal of Photogrammetry and Remote Sensing* 60 (3) (2006), pp.195-211.

ACKNOWLEDGEMENTS

The Belgian Science Policy Office is gratefully acknowledged for funding the work presented in this paper (SR/00/105). The authors also wish to thank the other partners of the MAMUD project, Dr. G. Buyuksalih of IMP-Bimtas and Prof. Dr. M. Crespi, Francesca Fratarcangelli and Francesca Pieralice of DITS – Area di Geodesia e Geomatica, Sapienza University Of Rome.

AUTOMATED SELECTION OF TERRESTRIAL IMAGES FROM SEQUENCES FOR THE TEXTURE MAPPING OF 3D CITY MODELS

Sébastien Bénitez and Caroline Baillard
SIRADEL, 3 allée Adolphe Bobierre CS 24343, 35043 Rennes, France
sbenitez@siradel.com

KEY WORDS: Building, Texture, Image, Sequences, Terrestrial, Automation.

ABSTRACT:
The final purpose of this study is to texture map existing 3D building models using calibrated images acquired with a terrestrial vehicle. This paper focuses on the preliminary step of automated selection of texture images from a sequence. Although not particularly complex, this step is particularly important for large-scale facade mapping where thousands of images might be available. Three methods inspired from well-know computer graphics techniques are compared: one is 2D-based and relies on the analysis of a 2D map; the two other methods use the information provided by a 3D vector database describing buildings. The 2D approach is satisfactory in most cases, but facades located behind low buildings cannot be textured. The 3D approaches provide more exhaustive wall textures. In particular, a wall-by-wall analysis based on 3D ray tracing is a good compromise to achieve a relevant selection whilst limiting computation.

1. INTRODUCTION

With the development of faster computers and more accurate sensors (cameras and lasers), the automatic and large-scale production of a virtual 3D world very close to ground truth has become realistic. Several research laboratories around the world have been working on this issue for some years. Früh and Zakhor have proposed a method for automatically producing 3D city models using a land-based mobile mapping system equipped with lasers and cameras; the laser points are registered with an existing Digital Elevation Model or vector map, then merged with aerial LIDAR data (Früh and Zakhor, 2003; Früh and Zakhor, 2004). At the French National Geographical Institute (IGN), the mobile mapping system Stereopolis has been designed for capturing various kinds of information in urban areas, including laser points and texture images of building facades (Bentrah *et al*., 2004). The CAOR laboratory from ENSMP has also been working on a mobile system named LARA-3D for the acquisition of 3D models in urban areas (Brun *et al.,* 2007; Goulette *et al.,* 2007), based on laser point clouds, a fish-eye camera, and possibly an external Digital Elevation Model. Recently, a number of private companies have commercialized their own mobile mapping systems for 3D city modeling, like StreetMapper or TITAN for instance (Hunter, 2009; Mrstik *et al*., 2009). The purpose of such systems is often the 3D modeling as well as the texture mapping of the 3D models.

In this study we are interested in texturing existing 3D building models by mapping terrestrial images onto the provided façade planes. As a part of the mapping strategy, one first needs to determine which images each façade can be seen from. It is particularly important for large-scale facade texture mapping where thousands of images can be available. Every single image can be relevant for the final texturing stage. There are few references on this issue. In (Pénard *et al.,* 2005) a 2D map is used to extract the main building facades and the corresponding images. All the images viewing at least a part of a façade are selected. In (Haala, 2004), a panoramic camera is used and a single image is sufficient to provide texture for many façades. Given a façade, the best view is the one providing the highest resolution. It is selected by analyzing the orientations and distances of the building facades in relation to the camera stations. In (Allène, 2008), a façade is represented by a mesh. Each face of the mesh is associated to one input view by minimizing an energy function combining the total number of texels representing the mesh in the images, and the color continuity between two neighbouring faces.

In our study, only two triangles per facade are available, and a façade texture generally consists of a mixture of 4 to 12 input views. The following mapping strategy has been chosen for texturing a given façade:

- Pre-selecting a set of relevant input images, from which the façade can be seen;
- Merging these images into a single texture image;
- Registering the texture image with the existing façade 3D model.

This paper only focuses on the first stage. The purpose of this operation is to select a set of potentially useful images based on purely geometrical criteria. The generation of a seamless texture image without occlusion artifacts will be handled within the second stage. Three possible approaches for the image pre-selection are presented and discussed. The first approach is similar to the one used in (Pénard *et al.,* 2005) and relies on the analysis of a 2D map. The two other methods use the information provided by a 3D vector database describing buildings. All methods are based on standard techniques commonly used in computer graphics for visibility computations, namely the ray-tracing and z-buffering techniques (Strasser, 1974). These two techniques have now been used for decades and are very well known in the computer graphics community. They can easily be optimized and accelerated via a hardware implementation.

This paper is organized as follows. Section 2 presents the test data set used for this study. Sections 3, 4 and 5 detail the three selection methods. The results and perspectives are discussed in section 6.

2. TEST DATASET

The test area is a part of the historical center of the city of Rennes in France. It is 1 km² wide and corresponds to the densest part of the city. Existing 3D building models were provided with an absolute accuracy around 1m. It contains 1475 buildings consisting of 11408 walls. The texture image database associated to the area was simulated via a virtual path created through the streets. A point was created every 5 meters along this path. Each point is associated to two cameras facing the left and the right sides of the path. The camera centers are located at 2.3 meters above the ground in order to simulate a vehicle height. The internal and external parameters of the cameras are approximately known. The path is about 4.9 kilometers long,

and it contains 990 points and 1980 cameras views (see Figure 1). It includes loops, self-intersections and close parallel roads. As a result a building wall can be seen from several locations within the path.

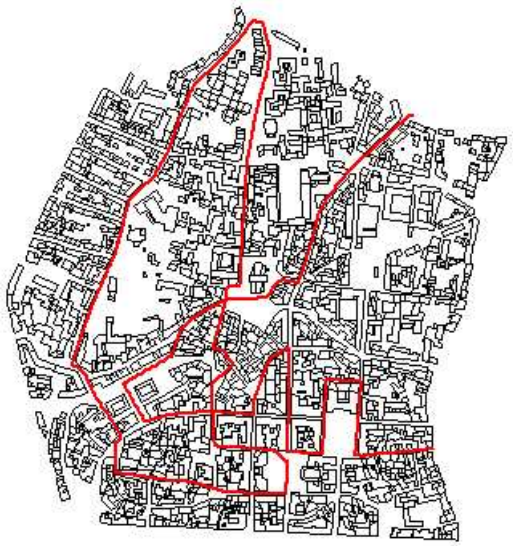


Figure 1. Test area, Rennes historical center. The virtual path is depicted in red.

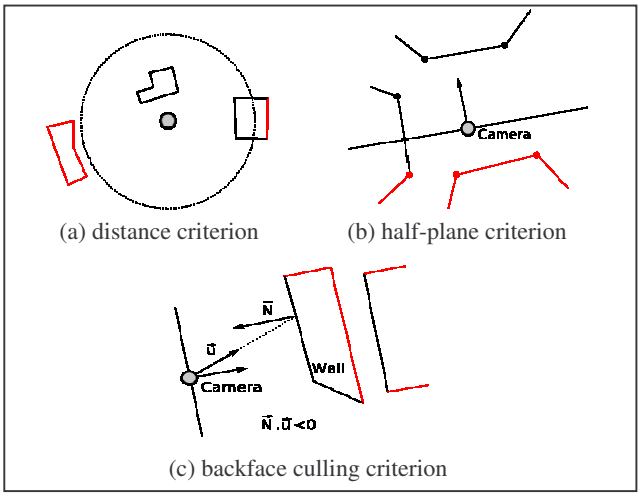
3. 2D RAY TRACING

3.1 Principle

The 2D approach is based on ray-tracing. Each camera is analyzed in turn. The walls are represented by 2D segments. For each camera a set of compatible wall segments is pre-selected using three criteria (see Figure 2):

- Distance criterion: the wall is located within a given distance from the camera center.
- Half-plane criterion: at least one point of the wall segment is located in the half-space in front of the camera
- Backface culling criterion: the wall is facing the camera.

The compatible wall segments define a set of candidate walls that might be visible from the current camera. An example of pre-selection is shown in Figure 10a-b.



(a) distance criterion (b) half-plane criterion

(c) backface culling criterion

Figure 2. The three criteria for the selection of candidate walls: (black=pre-selected walls, red=rejected walls)

The 2D-tracing technique is then applied to the candidate wall segments. First a beam of 2D lines is defined passing through the camera center point and regularly distributed within the field of view of the camera. Then the closest intersected candidate wall segment is selected as a visible wall. When all the cameras have been processed then each wall can be associated to the list of cameras that can view it.

3.2 Test results

The method was tested with various numbers of rays per camera. The distance threshold was arbitrarily set to 150m, distance above which the texture resolution is low enough to be discarded. The computing time includes reading and exporting steps. Numerical results are shown in Table 1. Between 10 and 13% of the walls are detected as visible by the process. Figure 3 shows the evolution of the wall number and the computing time with the number of rays. A qualitative example of selected walls can be found in Figure 10c.

Ray #	Total # of selected walls	Avg # of cameras per wall	Computing time
10	1176 (10.3%)	4.54	7s
50	1391 (12.1%)	4.95	11s
100	1450 (12.7%)	5.04	17s
500	1507 (13.2%)	5.14	50s

Table 1. Results of 2D ray tracing

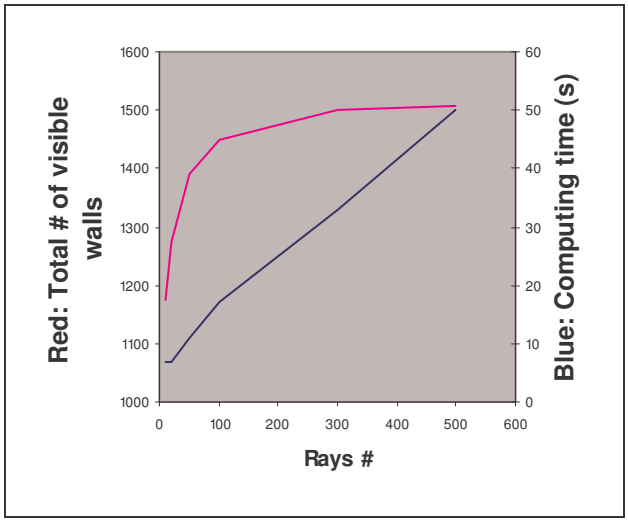
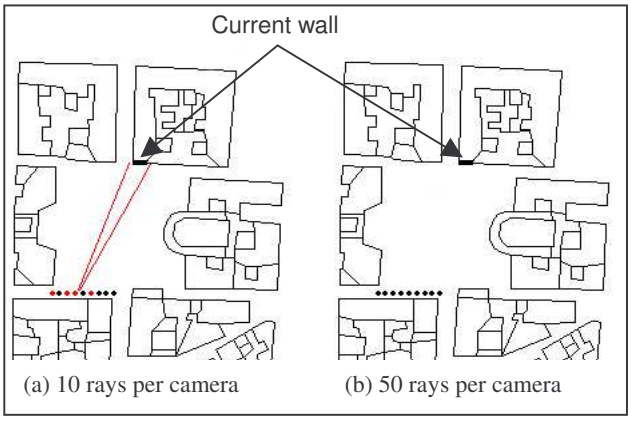


Figure 3 –Number of visible walls and computing time in relation to the number of rays

3.3 Discussion

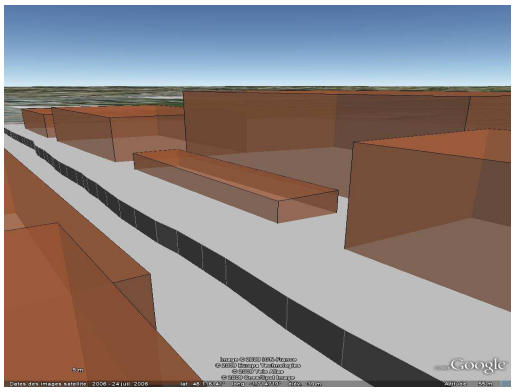
The variations in the number of selected walls come either from walls located far away of the camera, or from walls almost aligned with the camera center. When the number of rays is small, then many walls are located between two rays and are therefore not selected (see Figure 4). In our configuration, a number of rays around 100 seems to be a good compromise to get a maximum number of relevant images per building wall.

The main advantage of the 2D approach is the speed. It is also very simple and quick to implement. However it does not take building heights into account. Yet a low building (garage, shop, etc) may only mask the bottom part of higher buildings located behind it, especially if the camera is located on the top of a vehicle (see examples in Figure 5 and Figure 11a-c). Therefore it seems very important to make use of 3D information within the selection process.

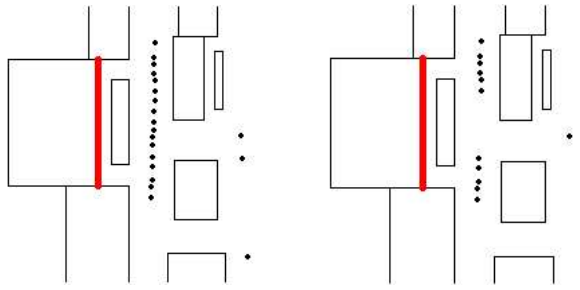


(a) 10 rays per camera (b) 50 rays per camera

Figure 4. Example of a "missed" wall. (a) 10 rays per camera: the current wall is "missed" by several cameras (red dots); (b) 50 rays per camera: the current wall is seen by all the cameras (black dots)



(a) 3D view



(b) Cameras seeing the red wall (c) Cameras selected with the 2D approach

Figure 5. Example of incomplete camera selection

4. 3D Z-BUFFERING

4.1 Principle

The second approach is 3D-based and relies on a z-buffer technique. Each camera is analysed in turn. A set of candidate walls is first associated to the current camera as in described in section 3, using distance, half-plane and backface culling criteria. The camera is then associated to a label image identifying the walls seen by the camera, and a depth image indicating the distance from the camera centre to the walls. Finally, after all the cameras have been processed, each wall can be associated to the list of cameras that can view it.

4.2 Test results

In order to reduce computing time, the distance and label images are sub-sampled at coarser resolutions. The tests were performed at a sampling resolution of 5, 10 and 20 pixels. They are shown in Table 2. It is important to note that the algorithm was not optimised and the graphical card not used. An example of depth image is shown in Figure 6.

Z-buffer resolution	Image size (hxw)	Total # of visible walls	Avg # of cam. per wall	Computing time
5 pixels	216x384	2310 (20.2%)	4.40	61min58s
10 pixels	108x192	2249 (19.7%)	4.41	52min36s
20 pixels	54x96	2186 (19.2%)	4.35	43min54s

Table 2. Results of 3D ray tracing

Figure 6. Example of a depth image

4.3 Discussion

Using this approach, 50% more walls can be textured. In particular, all facades located behind other buildings can now be textured, whereas they were discarded with the 2D approach. In the example of Figure 10d, the circled area shows an example of a high building visible from the current camera but only selected with the 3D approach. As the measures are very dense, even small walls, walls distant from the path or wall aligned with the path can theoretically be textured.

In return, many selected walls are only partly visible, and would actually have a very poor texture quality. It is important to introduce a contribution culling technique, in order to discard wall images inappropriate for texture mapping. In the current implementation, another drawback of the method is the computing time. Using a hardware implementation directly into the Graphical Processing Unit of the graphic card should solve this problem. A hierarchical z-buffer technique could also be investigated (Greene, 1993). Finally, the selection process must be entirely completed before being able to further process a façade, which might not be compatible with a large-scale production process.

5. 3D RAY TRACING

5.1 Principle

The last approach combines the main advantages of the two previous ones: speed and use of 3D information. It is a 3D extension of the 2D approach based on ray tracing. However, the analysis is performed wall-by-wall rather than camera-by-camera. Given a wall, a set of candidate cameras is selected using a method similar to the one described in section 3:

- The camera is located within a given distance from the wall (distance measured at closest point).
- The camera center point is located in the half-space in front of the wall.
- The camera is facing the wall plane.

In order to reduce computing time and improve texture quality, an additional criterion has been introduced: cameras that are almost aligned with the wall plane are discarded. A maximum threshold on the angle defined by the wall plane and the camera directions is introduced (see Figure 7). This filtering step is an extension of the backface culling criterion.

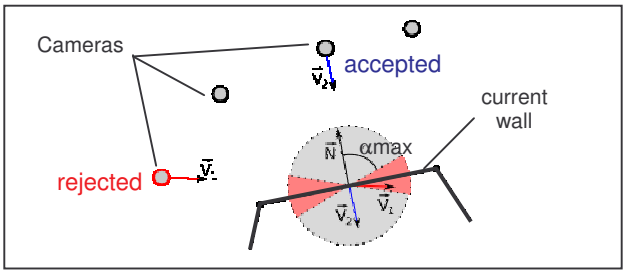


Figure 7. Angle criterion for the pre-selection of candidate cameras: the cameras with their direction vector in the red angular area are discarded

For each candidate camera a grid on the camera plane is defined. Each grid point defines a 3D ray passing through the camera center point. The 3D rays not intersecting the current wall are ignored. The remaining 3D rays are tested with respect to all the walls compatible with the camera (pre-selection method described in section 3): any ray intersecting a wall face closer than the current one is discarded. The candidate camera is finally selected as viewing the current wall, if at least one of the rays has not been discarded. The process is illustrated in Figure 8.

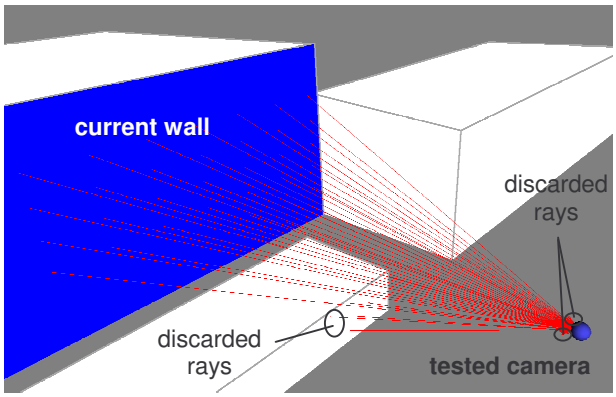


Figure 8. Principle of 3D ray tracing: the rays launched from the tested camera are discarded if they do not intersect the current wall (see rays on the extreme sides) or if they first intersect a closer wall (see rays on the left)

5.2 Test results

The method was tested with 10x10 and 20x20 rays per camera, with and without threshold on the angle during pre-selection. The threshold on angle was set to $\frac{3\pi}{8}$ radians when applied. The distance threshold was set to 150m (identical to 2D ray tracing). Numerical results are shown in Table 3. An example of selected walls is illustrated in Figure 10d.

Method	Total # of visible walls	Avg # of cam. per wall	Computing time
10x10 rays	1349 (11.8%)	4.36	3min51s
20x20 rays	1604 (14%)	4.49	11min45s
10x10 rays αmax = 3*Π/8	1032 (9%)	4.55	2min49s
20x20 rays αmax = 3*Π/8	1213 (10.6%)	4.56	8min25s

Table 3. Results of 3D ray tracing

5.3 Discussion

As expected from a 3D-based approach, the walls located at the background can be textured if they are high enough. Fewer texture images are selected with the 3D ray-tracing approach than with the z-buffer approach, but they generally have a better quality. It is not surprising as ray tracing is not a dense approach and most small wall textures are naturally discarded. In the example of Figure 10e, only the relevant facade of the high building located at the back of the block was selected as visible. Figure 11 shows another example of distant facade that can be textured only with a 3D approach.

The additional pre-selection criterion on angles removes sidelong walls, which are usually seen by few cameras. It improves the relevance of the selection by discarding walls with a poor texture resolution. Using 20x20 rays instead of 10x10 rays significantly increases the total number of visible walls, but further tests are needed in order to find out whether these additional walls can be textured with a good enough quality. Importantly, as each wall is processed in turn, the texturing stage can be performed without requiring the complete processing of the path.

The computing time is intermediate between 2D ray tracing and 3D z-buffering. In our implementation many calculations are redundant. A spatial division of the space could be performed in order to make use of object-space coherence and accelerate ray tracing (Glassner, 1984; Jevans and Wyvill, 1989).

6. CONCLUSION AND PERSPECTIVES

The 2D approach is satisfactory in most cases, and it is fast, simple and easy to implement. However any building located behind another cannot be textured.

The 3D approaches provide more exhaustive wall textures, including texture images for high building walls located at the back of lower buildings. The use of the 3D dimension makes the visibility estimation closer to ground truth, and the selection process more efficient. Although both ray-tracing and z-buffer techniques can be implemented very efficiently, the approach based on 3D ray tracing is a good compromise to achieve a relevant selection. It also seems important to prefer a wall-by-wall analysis, as further texturing stages can then be performed without requiring the complete processing of the path. The z-buffering technique could be considered if the resulting depth image is a valuable source of information in further steps.

The main constraint for the 3D approaches is obviously the availability of a 3D building database. Given a 2D map, the 3D information can be derived from a correlation-based or LIDAR Digital Elevation Model, or even from the analysis of architectural plans or building permits. In our opinion, a coarse 3D city model is sufficient to significantly improve the relevance of the texture selection.

We are now working on refining the selection with texture quality criteria rather than just visibility. The texture quality

depends on its resolution and varies with the distance from the camera to the facade.
Another possible evolution is to use additional 3D information to predict occlusions. A Digital Terrain Model could be used to predict hidden parts due to hills or embankments (case of a hill masking buildings facades located on the other side of a square for instance). If available, a complete 3D city model including vegetation and detailed building roofs would help better estimate the visibility of a given façade. More generally, an environment mask as in described in (Wang et al., 2002) could be introduced.
Another parameter to take into account is the uncertainty on the GPS/IMU data which introduces an uncertainty on the camera position and direction. In order to guarantee a complete selection, a simple solution would be to dilate each wall polygon by the maximal distance induced by the positioning uncertainty. In a similar way, the influence of the input 3D model accuracy should be investigated.
For this particular study, only synthetic data have been used. In the future we will be working on real data, and the influence of both the positioning error and the 3D model accuracy will be studied. Figure 9 gives an idea of what we would like to automatically achieve at a large scale. Note that the side facade located at the top right of the image cannot be textured if the image selection process is only 2D-based.

Figure 9 – 3D virtual view of the historical centre of Rennes

REFERENCES

Allène, C., Pons, J.P. and Keriven R., 2008. Seamless image-based texture atlases using multi-band blending. *Pattern Recognition, ICPR 2008.*

Bentrah, 0., Paparoditis, N., Pierrot-Deseilligny, M., 2004. Stereopolis : An Image Based Urban Environments Modelling System. In *International Symposium on Mobile Mapping Technology (MMT)*, Kunming, China, March 2004.

Brun, X., Deschaud, J.E. and Goulette, F., 2007, On-the-way City Mobile Mapping Using Laser Range Scanner and Fisheye Camera, In *International Symposium on Mobile Mapping Technology (MMT)*, Padua, Italy2007.

Früh, C., and Zakhor, A., 2003. Constructing 3D City Models by Merging Aerial and Ground Views. IEEE Computer Graphics and Applications, 23(6), Nov. 2003.

Früh, C. and Zakhor, A., 2004. An Automated Method for Large-Scale, Ground-Based City Model Acquisition. *International Journal of Computer Vision*, 60(1), pp. 5-24.

Glassner, A., 1984. Space subdivision for fast ray tracing. *IEEECG&A*, 4(10):15-22, Oct. 1984.

Goulette, F., Nashashibi, F., Abuhadrous, I., Ammoun, S. and Laurgeau, C., 2007. An Integrated On-board Laser Range Sensing System for On-the-way City and Road Modelling. In *ISPRS Comm. I Symposium,* Marne-la-Vallée, France, 2004.

Greene, N., Kasse, M., Miller, G., 1993. Hierarchical Z-buffer visibility. In *Proc. Of the 20th conf. On Computer graphics and interactive techniques,* Anaheim, CA, 1993.

Haala, N., 2004. On the refinement of urban models by terrestrial data collection. In *XXth ISPRS Congress, Vol. 35, Part B,* Istanbul, Turkey, 2004.

Hunter, G, 2009. Streetmapper mobile mapping system and applications in urban environments. In *ASPRS Annual Conference*, Baltimore, USA, 2009

Jevans, D. and Wyvill, B. Adaptive voxel subdivision for ray tracing. *Proc. Graphics Interface '89* , 164-172, June 1989.

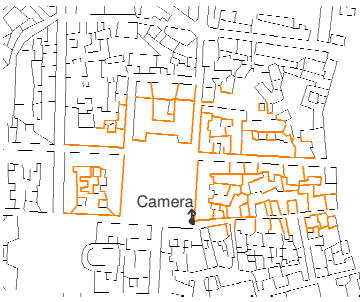
Mrstik, P., and Kusevic, K., 2009. Real Time 3D Fusion of Imagery and Mobile Lidar, *ASPRS Annual Conference*, Baltimore, USA, 2009.

Pénard, L., Paparoditis, N. and Pierrot-Deseilligny, M., 2005. 3D Building Facade Reconstruction under Mesh Form from Multiple Wide Angle Views, In *IAPRS vol. 36 (Part 5/W17),* 2005.

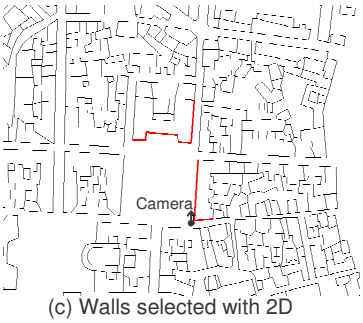
Strasser, W. *Schnelle kurven und Flächendarstellung auf graphishen Sichtgeräten*, Ph.D. Thesis D83, Technical University of Berlin, Germany, 1974

Wang X., Totaro S., Taillandier F., Hanson A. R. and Teller S., 2002. Recovering Facade Texture and Microstructure from Real-World Images. *Proc. of the 2nd International Workshop on Texture Analysis and Synthesis in conjunction with ECCV'02, pp. 145-149*, Copenhague, Denmark, June 2002.

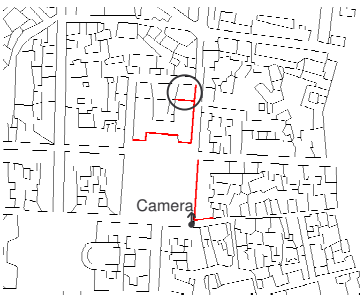
(a) Building heights
(blue=low, orange=high)



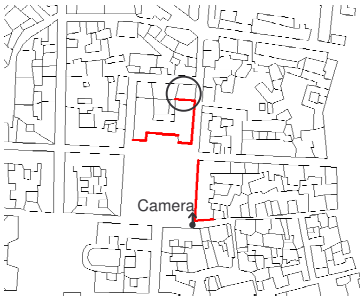
(b) Pre-selected building
walls (orange)



(c) Walls selected with 2D
ray tracing (red)



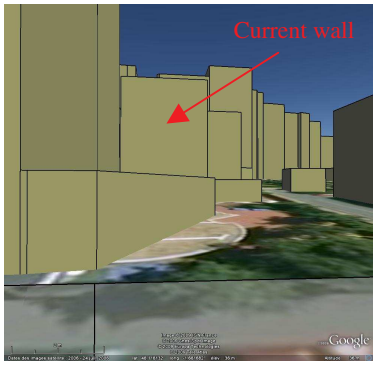
(d) Walls selected with 3D
z-buffers (red)



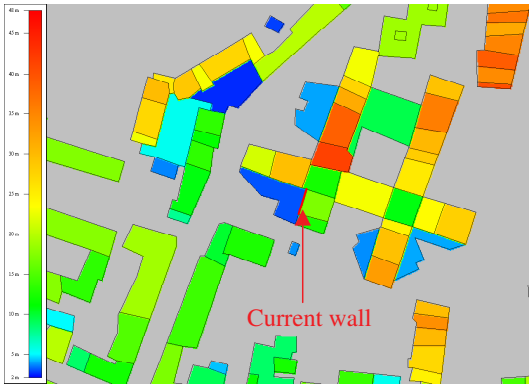
(e) Walls selected with 3D
ray tracing (red)

Figure 10. Results of the various selection methods

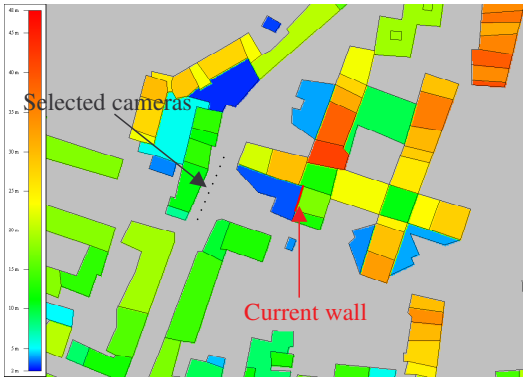
(a) Real view



(b) Virtual view of the input 3D model



(c) Result of the camera selection by 2D ray-tracing:
no camera was detected for the current wall



(d) Result of the camera selection by 3D ray-tracing:
8 cameras were found viewing the current wall.

Figure 11. Example of a low building masking a façade

CLASSIFICATION SYSTEM OF GIS-OBJECTS USING MULTI-SENSORIAL IMAGERY FOR NEAR-REALTIME DISASTER MANAGEMENT

Daniel Frey and Matthias Butenuth

Remote Sensing Technology
Technische Universität München
Arcisstr. 21, 80333 München, Germany
daniel.frey@bv.tum.de, matthias.butenuth@bv.tum.de

KEY WORDS: System, Classification, Statistics, Multisensor, Integration, GIS, Disaster

ABSTRACT:

In this paper, a near-realtime system for classification of GIS-objects is presented using multi-sensorial imagery. The system provides a framework for the integration of different kinds of imagery as well as any available data sources and spatial knowledge, which contributes information for the classification. The goal of the system is the assessment of infrastructure GIS-objects concerning their functionality. It enables the classification of infrastructure into different states as destroyed or intact after disasters such as floodings or earthquakes. The automatic approach generates an up-to-date map in order to support first aid in crisis scenarios. Probabilities are derived from the different input data using methods such as multispectral classification and fuzzy membership functions. The main core of the system is the combination of the probabilities to classify the individual GIS-object. The system can be run in a fully automatic or semi-automatic mode, where a human operator can edit intermediate results to ensure the required quality of the final results. In this paper, the performance of the system is demonstrated assessing road objects concerning their trafficability after flooding. By means of two test scenarios the efficiency and reliability of the system is shown. Concluding remarks are given at the end to point out further investigations.

1 INTRODUCTION

A significant increase of natural disasters such as floodings and earthquakes has been observed over the past decades (Kundzewicz et al., 2005). There is no doubt that the disasters' impact on the population has dramatically increased due to the growth of population and material assets. The regrettable death of people is accompanied by heavy economic damage, which leads to a long-term backslide of the regions hit by the disaster. This situation calls for the development of integrated strategies for preparedness and prevention of hazards, fast reaction in case of disasters, as well as damage documentation, planning and rebuilding of infrastructure after disasters. It is widely accepted in the scientific community that remote sensing can contribute significantly to all these components in different ways, in particular, due to the large coverage of remotely sensed imagery and its global availability.

However, time is the overall dominating factor once a disaster hits a particular region to support the fast reaction. This becomes manifest in several aspects: firstly, available satellites have to be selected and commanded immediately. Secondly, the acquired raw data has to be processed with specific signal processing algorithms to generate images suitable for interpretation, particularly for Synthetic Aperture Radar (SAR) images. Thirdly, the interpretation of multi-sensorial images, extraction of geometrically precise and semantically correct information as well as the production of (digital) maps need to be conducted in shortest timeframes to support crises management groups. While the first two aspects are strongly related to the optimization of communication processes and hardware capabilities, at least to a large extend, further research is needed concerning the third aspect: the fast, integrated, and geometrically and semantically correct interpretation of multi-sensorial images.

Remote sensing data was already used in order to monitor natural disasters in the year 1969 (Milfred et al., 1969). Particularly, in the case of flooding a lot of studies are carried out to infer information as flood masks from remote sensing data (Sanyal and Lu, 2004). The flooded areas can be derived from optical images (Van der Sande et al., 2003) as well as from radar images (Martinis et al., 2009) via classification approaches. Zwenzner (Zwenzner and Vogt, 2008) estimates further flood parameter as water depth using flood masks and a very high resolution digital elevation model. Combining this results with GIS data leads to an additional benefit of information and simplifies the decision making (Brivio et al., 2002, Townsend and Walsh, 1998). The combination of the GIS and remote sensing data is often carried out by overlaying the different data sources. But, there are only few approaches which use the raster data from imagery to assess the given GIS data. In (Gerke et al., 2004, Gerke and Heipke, 2008) an approach for automatic quality assessment of existing geospatial linear objects is presented. The objects are assessed using automatically extracted roads from the images (Wiedemann and Ebner, 2000, Hinz and Wiedemann, 2004). However, in case of natural disasters the original roads are destroyed or occluded and, therefore, it is not possible to extract them using the original methods. Hence, new approaches have to be developed which assesses damaged and occluded objects, too. The integration and exploitation of different data sources, e.g. vector and image data, was discussed in several other contributions (Baltsavias, 2004, Butenuth et al., 2007). However, there is a lack of methods which assess the GIS data concerning its functionality using imagery (Morain and Kraft, 2003).

In this paper, a classification system using remote sensing data and additionally available information is developed to assess GIS-objects. The main goal of the system is the automatic classification and evaluation of infrastructure objects, for example the trafficability of the road network after natural disasters. However, the presented system can be transferred to other scenarios, such as changes in vegetation, because its design is modular. A focus is the integrated utilization of any available information, which is important to ease and speed up the classification process with the aim to derive complete and reliable results (Reinartz et al., 2003, Frey and Butenuth, 2009). In comparison to the manual interpretation of images the presented systems is very efficient,

which is essential in crisis scenarios. Depending on the type and complexity of the input data, the system can be run in a fully automatic or semi-automatic mode, where a human operator can edit intermediate results to ensure the required quality of the final results.

Section 2 describes the generic near-realtime classification system with the objective to classify and evaluate objects using remote sensing and other available data. In Section 3 the system is applied to road objects in case of natural disasters. Two test scenarios of flooded areas are used to verify the system. By means of manually generated reference data, the applicability and efficiency of the system is evaluated in Section 4. Finally, further investigations in future work are pointed out.

2 CLASSIFICATION SYSTEM

The goal of the developed classification system is the assessment of GIS-objects using up-to-date remote sensing data. The system is designed in a general and modular way to provide the opportunity to label GIS-objects into different states. Typical states describe the functionality of infrastructure objects as roads or buildings. The generic system embeds different kinds of image data: multi-sensor as well as multi-temporal data. Additionally, any kinds of available data sources and spatial knowledge, which contributes information for the assessment, can be embedded. Typical examples are digital elevation models (DEM) and further GIS information, e.g. land cover or waterways. The minimum requirement of the system are the objects to be assessed and one up-to-date image which provides the information for the assessment.

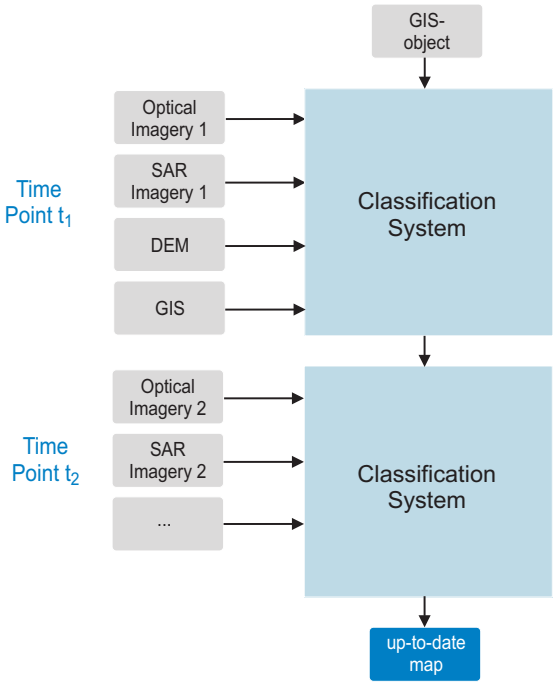


Figure 1: Classification system

The classification system depicted in Figure 1 can be subdivided into different components. Starting point are the GIS-objects to be assessed. Secondly, the input data as imagery or digital elevation models which contribute the information for the assessment. In the following this information is called *data*. Thirdly, the classification system by itself and, finally, a resulting up-to-date map.

The fusion of multi-sensor images is an important issue, because the corregistration between optical and radar images is still a current research topic (Pohl and Van Genderen, 1998). Methods such as mutual information can be applied for the system (Inglada and Giros, 2004). The system has to deal with multi-temporal images having the possibility to derive important information on time. This leads to an even more complex corregistration process. Change detection algorithms can provide information about the variation of assessed objects. In this article the temporal factor is neglected, but will be an essential part in future research.

The main core of the system represents the classification. The goal is to classify each object into a different state S_i. For each object probabilities are derived belonging to a certain state. The methods estimating the probabilities depends on the data: typical examples are multispectral classification or fuzzy membership functions (Figure 2).

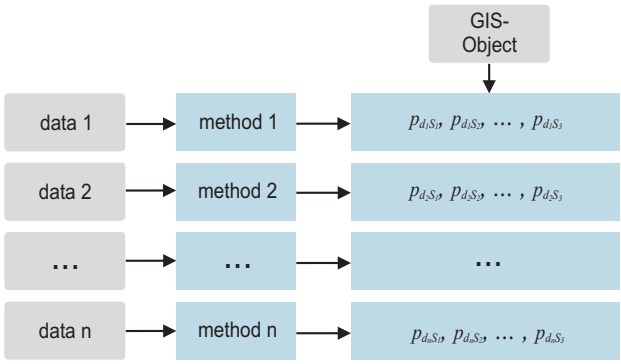


Figure 2: Derivation of probabilities from data using various methods

Beside the derivation of the individual probabilities from each data source the combination plays a decisive role:

$$\begin{aligned} p_{S_1} &= p_{d_1,S_1} \otimes p_{d_2,S_1} \otimes \cdots \otimes p_{d_n,S_1} \\ p_{S_2} &= p_{d_1,S_2} \otimes p_{d_2,S_2} \otimes \cdots \otimes p_{d_n,S_2} \\ &\vdots \\ p_{S_i} &= p_{d_1,S_i} \otimes p_{d_2,S_i} \otimes \cdots \otimes p_{d_n,S_i}. \end{aligned} \tag{1}$$

The variable p_{d_n,S_i} denotes the probability that the state S_i occurs given data d_n. The indices i and n describe the number of available states and data, respectively. The result p_{S_i} shows the probability that a GIS-object belongs to the state S_i. For each type of data weights w_n can be introduced in order to cope with the different influence of information content. Hence, Equation 1 for one state i leads to:

$$p_{S_i} = w_1 \cdot p_{d_1,S_i} \otimes \cdots \otimes w_n \cdot p_{d_n,S_i}. \tag{2}$$

Finally, the object is assigned to the state S_i with the largest probability p_{S_i}. A basic characteristic of the whole system is the combination at the probability level in order to remain flexible concerning the available data.

3 MODEL FOR ROAD OBJECTS

After describing the generic system, a model is shown which assesses linear objects as roads after flooding. However, this model is transferable to other linear objects like railways and further

natural disasters such as avalanches, landslides or earthquakes. In case of natural disasters the GIS-object can be divided into the state *intact/usable* or *not intact/destroyed*. Furthermore, a state between these extrema is possible. Hence, a third state *possibly not intact/destroyed* is introduced, if the automatic approach can not provide a reliable decision. In order to assess roads after a flood disaster following states can be used:

- *trafficable*
- *flooded*
- *possibly flooded*

For every available data source the probability for each state has to be derived. The methods which are employed to the different data are shown in the following section.

3.1 Methods

A multispectral classification is accomplished in order to derive different classes from the input imagery. The goal is to assess each linear object individually without taking adjacent linear objects into account, because such kind of topological knowledge about the connectivity of a road network is no more valid in case of road networks hit by a natural disaster. Every linear object is a polyline, which consists of several line segments. A line segment is a straight line, which can be defined with two points. Every line segment is assigned to a class using an segment-based multispectral classification. To this end, a buffer is defined around each line to investigate the radiometric image information. In many cases additional information as the width of the line object can be used in order to generate the size of the buffer region.

For the multispectral classification various classes have to be defined depending on the underlying imagery in order to classify the road segments into the three states *trafficable*, *flooded* and *possibly flooded*. In case of optical imagery the classes road, water, forest and clouds are convenient, because the class road corresponds to the state *trafficable*, the class water to *flooded* and the classes forest and clouds describe occlusions and therefore belong to the state *possibly flooded*. If radar images are available the class clouds can be neglected. Beside the assignment to a class each individual line segment consists of a probability belonging to a class ω_i, which is derived from the k-sigma error ellipsoid. The probability can be formulated as $p_{\omega_i}(\vec{g})$, whereas $\vec{g}$ defines the gray values. The length of the vector is equivalent to the number of channels.

Beside the imagery additional information such as digital elevation models or GIS data can be integrated in the system. The methods to derive probabilities depend on the data. One method are membership functions of fuzzy sets (Zadeh, 1965). Membership functions do not describe the likelihood of some event, but they only characterize a degree of truth in vaguely defined sets. Since it is often difficult to derive sound probabilities from GIS data, membership functions provide an opportunity to infer confidence values. To emphasize the distinction the membership function is labeled as μ instead of p.

The membership functions $\mu_t(a), \mu_f(a)$ are introduced if a digital elevation model is given. The function $\mu_t(a)$ denote the belonging to the state *trafficable* t depending on the altitude a. Similarly $\mu_f(a)$ represents the state *flooded* f. Both functions are depicted in Figure 3. There are two thresholds a_1 and a_2 which determine the height of very likely flooded or trafficable areas, respectively. The current water level lies between these thresholds, which can be calculated by

$$\begin{aligned} a_1 &= l_l - b_1 \\ a_2 &= l_h + b_2, \end{aligned} \tag{3}$$

in which l_l is the lowest and l_h is the highest water level in the scene. In order to involve variations due to flows and barriers additional buffers b_1, b_2 are added.

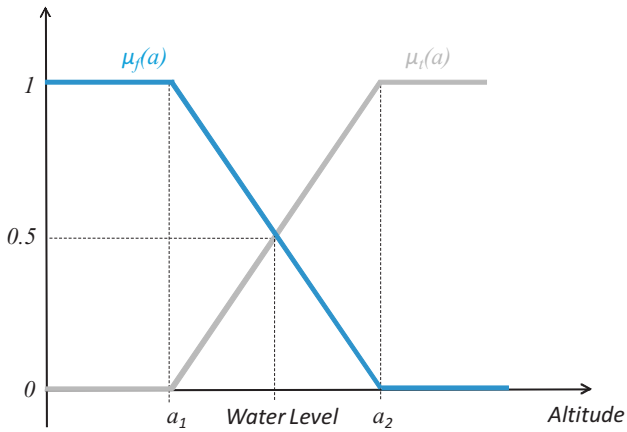


Figure 3: Membership functions for flooded roads and trafficable roads derived from DEM

3.2 Combination of Probabilities

The core of the classification system is to combine probabilities resulting from a multispectral classification with the degree of truth of membership functions. In this section, an example is shown which combines the derived probabilities from optical images with membership functions inferred from a digital elevation model. By means of multispectral classification for each class (water w, road r, forest o, cloud c) the corresponding probability p_{ω_i} for $i = \{w, r, o, c\}$ can be derived. On the other side, the membership function provide the degree of truth $\mu_t(a)$ and $\mu_f(a)$. Utilizing the knowledge that roads higher than a_2 are definitely trafficable and roads lower than a_1 are very likely flooded a case differentiation is carried out:

$$\mu_f(\vec{g}, a) = \begin{cases} \mu_f(a) = 1 & a \leq a_1 \\ \mu_f(a) \cdot p_{\omega_w}(\vec{g}) & a_1 < a < a_2 \\ \mu_f(a) = 0 & a \geq a_2 \end{cases} \tag{4}$$

$$\mu_t(\vec{g}, a) = \begin{cases} \mu_t(a) = 0 & a \leq a_1 \\ \mu_t(a) \cdot p_{\omega_r}(\vec{g}) & a_1 < a < a_2 \\ \mu_t(a) = 1 & a \geq a_2. \end{cases} \tag{5}$$

Variable a denotes the height of a road object. The road is assigned to the state *flooded* S_F if the degree of truth $\mu_f(\vec{g}, a)$ exceeds an threshold t_1, which can be pre-estimated via the standard deviation of the likelihood function resulting from the training data for water. The road is assigned to the state *possibly flooded* S_{PF}, if $\mu_f(\vec{g}, a)$ is less than t_1. The probability $\mu_t(\vec{g}, a)$ is treated in an analogous manner. The road is assigned to the state *trafficable* S_T if $\mu_t(\vec{g}, a)$ exceeds a pre-determined threshold t_2. Otherwise, the road is again assigned to the state *possibly flooded* S_{PF}. The road segments which are classified as forest ω_o or clouds ω_c are assigned to the states in the following way:

$$\begin{aligned} a < a_1 \quad &\Rightarrow \textit{ flooded } S_F \\ a_1 < a < a_2 \quad &\Rightarrow \textit{ possibly flooded } S_{PF} \\ a > a_2 \quad &\Rightarrow \textit{ trafficable } S_T \end{aligned} \tag{6}$$

In Figure 4 a schematic overview of the used classification system is depicted. A multispectral classification is carried out to assign the road objects to the different classes. The results of the multispectral classification combined with the membership function leads to the assignment of the road objects to the different states.

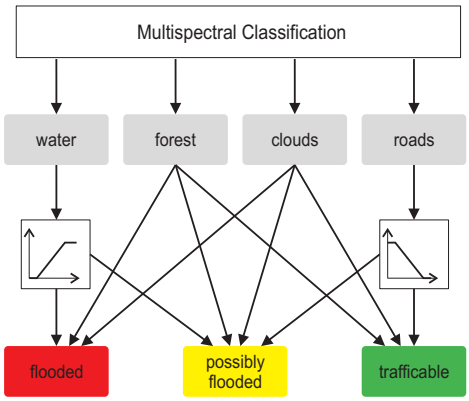


Figure 4: Schematic overview of the classification system

4 RESULTS AND EVALUATION

The presented system has been exemplarily tested with two scenarios representing flood disasters. In both cases roads are assessed concerning their trafficability. The first scenario is the Elbe flood in the year 2002 near Dessau, Germany. Three different data sources are used for the assessment: Firstly, an IKONOS-Image with four channels (red, green, blue and infrared), cf. Figure 5. The ground-sampling distance of the panchromatic channel is 1 meter and the color-channels is 4 meter. As second source a digital elevation model with a resolution of 10 meters is used. Finally, the objects to be assessed are taken form the ATKIS (German Official Topographic Cartographic Information System) database. The test scene covers an area of 33 km^2, which contains 5484 line segments. In the following investigations only the road objects are studied.

The second study area is located in Gloucesterhire Region in Southeast England. In July 2007 the record flood level at Tewkesbury was measured. During the flooding a TerraSAR-X scene in StripMap mode with a spatial resolution of 3 meter was acquired. The polarization is HH, which is more efficient than HV or VV to distinguish flooded areas (Henry et al., 2003). The test scene covers an area of of 9,5 km^2. Additionally, linear membership functions from the original rivers are derived and an automatically extracted flood mask is used. As GIS-objects 522 roads from OpenStreetMap are assessed.

The test scenarios are very appropriate to test the classification system due to their diverse global context and the different kinds of roads. The roads vary from paths to highways. Both test scenarios are evaluated using manually derived reference data. The availability of reference data describing the real status of roads during the flooding is very difficult caused by the fast changes of the water level and the accessibility of the roads. One possibility is to derive the reference data from the image itself, which is done for the Elbe scenario. This kind of reference data does not describe the ground truth, but the information which is possible to get from the studied image. In the case of the Gloucesterhire scenario high resolution airborne image with a resolution of 20 cm are available. This imagery which was acquired half a day later than the studied TerraSAR-X scene was used to infer the exact ground truth. To draw conclusions from the following results, it is important to consider the kind of used reference data.

The result of the Elbe scene is visualized in Figure 5. The red lines refer to flooded roads, green lines to trafficable roads and the yellow lines point out, that no decision is possible by the automatic system. In Figure 6 a detail of the original IKONOS image and the assessed roads is shown.

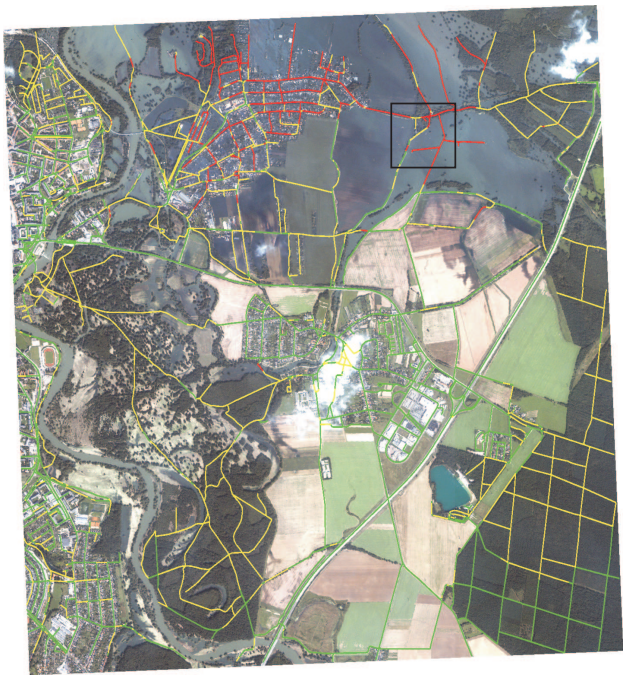


Figure 5: Automatic assessment of roads using the classification system: flooded roads (red), trafficable roads (green) and possibly flooded roads (yellow)

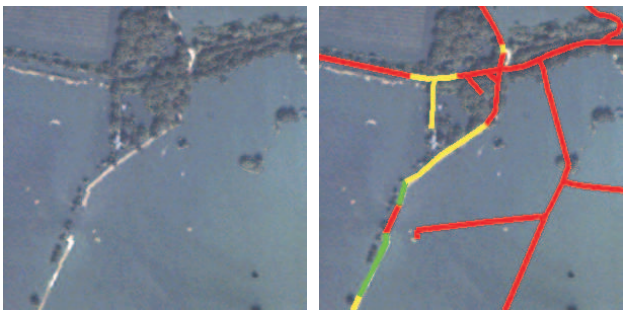


Figure 6: Detail of original and assessed IKONOS scene

Comparing the result with the manually generated reference leads to the numerical results shown in Table 1. ”Correct assignment” means that the manually generated classification is identical with the automatic approach. In the case of ”Manuel control necessary” the automatic approach leads to the state *possibly flooded* whereas the manual classification assigns the line segments to *flooded* or *trafficable*. The other way around denotes the expression ”Possibly correct assignment”. ”Wrong assignment”’ means that one approach classifies the line segment to *flooded* and the other to *trafficable*. With the current implementation of the system the approach achieves a correct assignment for 78% of the road objects. Only a very small value of false assignments is obtained. This result is deteriorated due to the 5% of ”Possibly wrong assignments”. Less than 1/5 of all road segments (17%)

should be controlled manually in order to reach a correctness of 95%.

Possible assignment	Result
Correct assignment	76.99%
Manual control necessary	17.87%
Possibly correct assignment	4.96%
Wrong assignment	0.18%

Table 1: Results Scenario: Elbe

The results are obtained with the threshold parameters $t_1 = 0.5$ and $t_2 = 0.001$. The variations of the parameters are depicted in Figure 7. The parameters are responsible for the amount of road segments which are assigned to the state *possibly flooded* on condition that they are classified to the classes water or road. The decrease of "Wrong assignment" comes along with the decrease of "Correct assignments" and an increase of manual control.

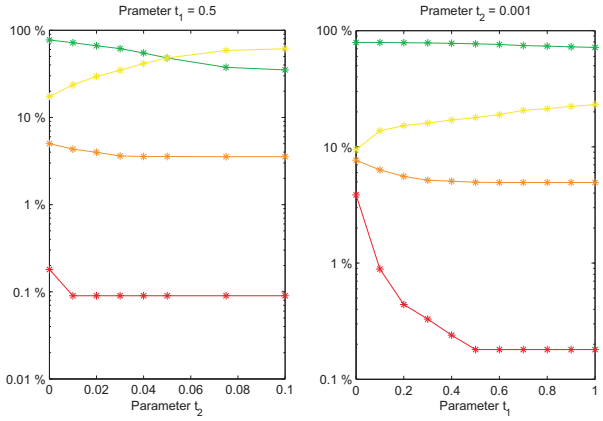


Figure 7: Results dependent on parameter t_1 and t_2 (red = Wrong assignment, orange = Possibly correct assignment, yellow = Manual control necessary, green = Correct assignment)

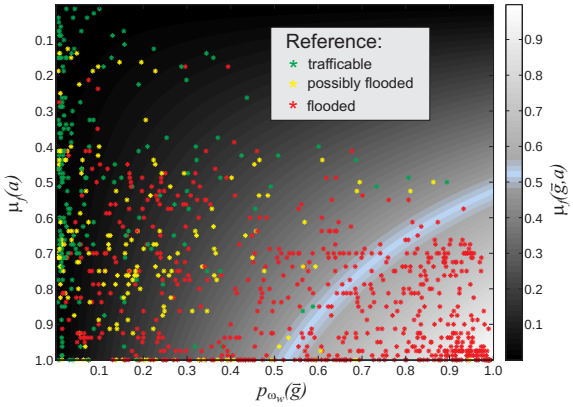


Figure 8: Combination of probabilities and impact of the parameter t_1

In Figure 8 the combination of the probabilities $\mu_f(a)$ and $p_{\omega_w}(\vec{g})$ is shown. The grayscale bar indicates the combined probability $\mu_f(\vec{g}, a)$. Every star defines a road segment assigned to the class water by multispectral classification, the color shows the state assigned in the reference. Many road segments which are assigned to the state *trafficable* in the reference are wrongly classified by the system to the class water. The reason is the high standard deviation of the probability densitiy function for the class road and, therefore, the overlapping of the class road and water. Road segments in urban areas occluded by shadows are responsible for this effect. The threshold t_1 is depicted in blue which devide the assignment of the roads to the state *flooded* and *possible flooded* (Figure 8). Shifting this parameter leads to the results illustrated on the right plot in Figure 7. Furthermore, the improvement of the combined probability is shown in Figure 8. If only one probability is available, the threshold t_1 would be depicted as a straight horizontal or vertical line. The total required time to generate the manual reference is about three hours. Compared to the time needed for the automatic classification (less than one minute) points out the efficiency of the approach.

The results of the second test scenario are depicted in Figure 9. A detail of the original TerraSAR-X scene and the assessed road segments is shown in Figure 10.

Figure 9: Automatic assessment of roads using the classification system: flooded roads (red), trafficable roads (green) and possibly flooded roads (yellow)

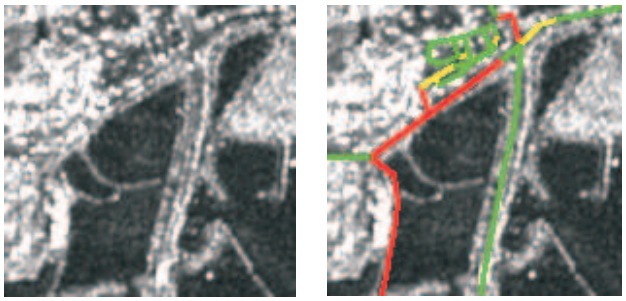


Figure 10: Detail of original and assessed TerraSAR-X scene

In the second test scenario the real ground truth is available. Hence, the assignment *possibly flooded* is not existing in the reference data. The comparison with the automatic classification system leads to the result shown in Table 2. After controlling 5% manually, altogether over 86% are correctly assigned. The value of 14% of wrong assignment is caused by mainly two reasons: Firstly, the resolution of the StripMap mode hardly enables to

detect flooded roads in urban areas. Secondly, the geometric accuracy of the used OpenStreeMap road objects are in many cases not accurate enough for a correct assignment.

Possible assignment	Result
Correct assignment	81.22%
Manual control necessary	4.60%
Wrong assignment	14.18%

Table 2: Results Scenario: Gloucesterhire

CONCLUSIONS

This article presents a classification system to assess GIS-objects concerning their functionality. The system is evaluated by means of two test scenarios with the goal to derive the trafficability of roads during a flooding. Both test scenarios show the good performance and especially the efficiency of this approach. In future work, the whole system will be evaluated using real ground truth to identify the reliability in disaster scenarios. Moreover, the additional benefit combining different image data types such as optical and radar will be part of further study. Currently, the combination of the probabilities is accomplished with a simple multiplication. It has to be investigated, if the combination of different probabilities could be realized better using a Dampster-Shafer framework. In addition, future work comprises the development of multi-temporal models to better exploit different image acquisition times including different data types. A further point is the preprocessing of the used GIS-objects to impove the spatial accuracy of the used infrastructure objects.

ACKNOWLEDGEMENTS

This work is part of the IGSSE project "SafeEarth" funded by the Excellence Initiative of the German federal and state governments, and part of the project "DeSecure". The author would like to thank the Federal Agency for Cartography and Geodesy Sachsen-Anhalt to provide the DEM and the ATKIS road data.

REFERENCES

Baltsavias, E., 2004. Object extraction and revision by image analysis using existing geodata and knowledge: current status and steps towards operational systems. ISPRS Journal of Photogrammetry and Remote Sensing 58(3-4), pp. 129–151.

Brivio, P., Colombo, R., Maggi, M. and Tomasoni, R., 2002. Integration of remote sensing data and GIS for accurate mapping of flooded areas. International Journal of Remote Sensing 23(3), pp. 429–441.

Butenuth, M., Gösseln, G., Tiedge, M., Heipke, C., Lipeck, U. and Sester, M., 2007. Integration of heterogeneous geospatial data in a federated database. ISPRS Journal of Photogrammetry and Remote Sensing 62(5), pp. 328–346.

Frey, D. and Butenuth, M., 2009. Analysis of road networks after flood disasters using multi-sensorial remote sensing techniques. Publikationen der Deutschen Gesellschaft für Photogrammetrie, Fernerkundung und Geoinformation 18, pp. 69 – 77.

Gerke, M. and Heipke, C., 2008. Image-based quality assessment of road databases. International Journal of Geographical Information Science 22(8), pp. 871–894.

Gerke, M., Butenuth, M., Heipke, C. and Willrich, F., 2004. Graph-supported verification of road databases. ISPRS Journal of Photogrammetry and Remote Sensing 58(3-4), pp. 152 – 165.

Henry, J., Chastanet, P., Fellah, K. and Desnos, Y., 2003. ENVISAT multipolarised ASAR data for flood mapping. Proceedings of Geoscience and Remote Sensing Symposium, IGARSS 2, pp. 1136–1138.

Hinz, S. and Wiedemann, C., 2004. Increasing efficiency of road extraction by self-diagnosis. Photogrammetric Engineering and Remote Sensing 70(12), pp. 1457–1464.

Inglada, J. and Giros, A., 2004. On the possibility of automatic multisensor image registration. IEEE Transactions on Geoscience and Remote Sensing 42(10), pp. 2104–2120.

Kundzewicz, Z., Ulbrich, U., Brücher, T., Graczyk, D., Krüger, A., Leckebusch, G., Menzel, L., Pińskwar, I., Radziejewski, M. and Szwed, M., 2005. Summer floods in central europe - climate change track? Natural Hazards 36(1), pp. 165–189.

Martinis, S., Twele, A. and Voigt, S., 2009. Towards operational near real-time flood detection using a split-based automatic thresholding procedure on high resolution TerraSAR-X data. Natural Hazards and Earth System Science 9(2), pp. 303–314.

Milfred, C., Parker, D. and Lee, G., 1969. Remote sensing for resource management and flood plain delineation. 24th Midwestern States Flood Control and Water Resources Conference.

Morain, S. and Kraft, W., 2003. Transportation lifelines and hazards: Overview of remote sensing products and results. Proceedings of Remote Sensing for Transportation 29, pp. 39 – 46.

Pohl, C. and Van Genderen, J., 1998. Multisensor image fusion in remote sensing: concepts, methods and applications. International Journal of Remote Sensing 19, pp. 823–854.

Reinartz, P., Voigt, S., Peinado, O., Mehl, H. and Schroeder, M., 2003. Remote sensing to support a crisis information system: Mozambique rapid flood mapping system, river elbe flood: Germany 2002. Proceedings of Remote Sensing of Environment pp. 10–14.

Sanyal, J. and Lu, X., 2004. Application of remote sensing in flood management with special reference to monsoon Asia: a review. Natural Hazards 33(2), pp. 283–301.

Townsend, P. and Walsh, S., 1998. Modeling floodplain inundation using an integrated GIS with radar and optical remote sensing. Geomorphology 21(3-4), pp. 295–312.

Van der Sande, C., De Jong, S. and De Roo, A., 2003. A segmentation and classification approach of IKONOS-2 imagery for land cover mapping to assist flood risk and flood damage assessment. International Journal of Applied Earth Observations and Geoinformation 4(3), pp. 217–229.

Wiedemann, C. and Ebner, H., 2000. Automatic completion and evaluation of road networks. International Archives of Photogrammetry and Remote Sensing 33(B3/2; PART 3), pp. 979–986.

Zadeh, L., 1965. Fuzzy sets. Information and Control 8(3), pp. 338–353.

Zwenzner, H. and Vogt, S., 2008. Improved estimation of flood parameters by combining space based SAR data with very high resolution digital elevation data. Hydrology and Earth System Sciences Discussions 5(5), pp. 2951 – 2973.

AN APPROACH FOR NAVIGATION IN 3D MODELS ON MOBILE DEVICES

Wen Jiang [a,c,] *, Wu Yuguo[b], Wang Fan [a]

[a] Institute of Surveying and Mapping, Information Engineering University, No.66, Longhai Road, Zhengzhou, R.P.China
[b] Zhengzhou Teachers college, Zhengzhou, R.P.China
[c] 78155 Troops, No.16, Shuxing Road, Chengdu, R.P.China
kissfro9642@sina.com
chxy_wenjiang37@yahoo.cn

Commission VI, WG VI/4

KEY WORDS: Navigation assistant, Mobile devices, Pyramid model, Quad-tree structure, Multi-resolution

ABSTRACT:

Traditional navigation visualization utilizes two-dimensional digital maps for road guidance, and with the advances in visualization technique, algorithms, and computer hardware, it offer an opportunity of applications for mobile users in 3D virtual environment. The main challenge comes from how to efficiently provide up-to-data location-specific data and navigation services. For the real 3D world usually contains a lot of details and represents a huge amount of datasets, so it is a difficult to visualize the complex virtual 3D scenes and navigate in them on mobile devices. To solve the problem, this paper proposed a novel approach that is based on geographic web services and the servers dynamically generate the 3D scenes in terms of the navigation commands and then send the resulting as video-encoded image stream to the mobile client. In order to enhance the efficiency of 3D scenes rendering, those virtual 3D models' datasets were prepared and organized in an offline process. The approach allows us to provide interactivity for complex virtual 3D scenes on resource and bandwidth limited mobile devices.

1. INTRODUCTION

Traditional navigation and trip planning is two-dimensional map display mode, and user can only accept limited information organization, poor presentation, and lack of interaction. On the contrary, applying 3D techniques for realistic visualizations into navigation fields provides new or better solutions that are hardly solved by 2D means, the advantages over 2D case are as follows: 1) easy navigation of the information space allowing better user interaction with the virtual objects and user can understand the displayed data through the existence of visual metaphors better. In contrast, 2D map reading is a skill which requires specific training; 2) the capability to display more data at one time, because each location on a 2D map is shown in the same scale, and users need to change scales in order to switch from viewing local details to overviews. The perspective view, on the other hand, has the inherent capability of combining different scales into one scene by dedicating a larger amount of the screen to the immediate surrounding while at the same time showing the entire route in an overview.

Mobile devices, like personal digital assistants (PDAs), mobile phones, Palm Pilots, or Pocket PCs, have made undreamed progress in computing power, function of displaying and input options a few years ago. Combined with a Global Positioning System (GPS) receiver, the mobile devices offers an opportunity to interact with a map display showing the current location and orientation.

Therefore, focused on how to represent 3D environment which support navigation on mobile devices, this paper presents a client-server solution for accessing virtual 3D scenes for navigation. The approach is based on 3D modelling techniques in which a full 3D model is generated on sever and sends the resulting as video-encoded image stream to the mobile client. The solution can be decomposed in three steps. The first one is pre-processing. The main purpose is preparing data offline. The second one is 3D scenes rendering. The user controls the client by navigation command sketches drawn directly on the view-plane and the sketches are sent to the server, then the server interprets these sketches in terms of navigation commands, and generates the 3D panorama scenes. The last one is sends the results as image sequences to the mobile client.

2. RELATED WORKS

Mobile applications of virtual 3D scenes represent a major and complex research challenge and bottlenecks due to limited bandwidth and graphic capabilities, restricted interaction capabilities, data standardizations and distribution techniques, and digital rights issues. The main challenge here is how to render the 3D scenes and models as to usable navigation task within the 3D environment because there lack of an efficient 3D engine and suitable 3D model that would allow such development and field experiments. And the other challenge for the navigation domain comes from how to maintain real-time update rates in loading and unloading large, complex datasets. In fact, 3D space data obtain from the natural environment is

* Institute of Surveying and Mapping, Information Engineering University, No.66, Longhai Road, Zhengzhou, R.P.China. 450052. E-mail: chxy_wenjiang37@yahoo.cn; kissfro9642@sina.com. Tel: 86-13017690807.

usually very huge and mobile platforms have limited computation resource (CPU power, memory, storage, and wireless network speed). There are several techniques have been proposed to visualize, navigate, interact, and query database systems in virtual environment.

2.1 3D Rendering techniques

Early studies on 3D maps often attempted to use mobile devices with direct model view software. In 3D computer graphics, numerous rendering techniques are available to cope with complex virtual environment, including discrete and continuous multi-resolution geometry and texture representations, view-frustum culling, occlusion culling, imposter techniques, and scene-graph optimizations (Akine-Möller and Haines 2002). Visualizations of virtual 3D city models and large terrain require an efficient management of large-scale texture data, such as images of building facades, aerial photography pictures of the terrain, and level-of-details (LOD) management for hierarchy of mesh refinement operations for large heterogeneous 3D object collections. Although these rendering techniques enable real-time rendering of complex 3D scene, they still cannot be rendered on mobile devices due to limited computational resources and power.

In order to efficient mobile 3D rendering, numerous techniques have been approached. Royan et al (2003) describe client-server architecture for mobile 3D virtual city visualizations based on a progressive and hierarchical representation for 3D geo-virtual environments. In his approach, the server firstly pre-computes multi-resolution representations of terrain models and building models, and then sends these data about visible areas to the mobile clients progressively. However, this method need clients implement rendering task dynamically and it is difficult to mobile devices due to the broad variety of hardware and software solutions for mobile 3D graphics (e.g. OpenGLES, Mobile 3D Graphics API for J2ME) (J. Döllner, B. Hagedorn and S. Schmidt 2006).

Another principle solution consists in server-side 3D rendering and the progressive, compressed transmission of image sequences. Cheng et al. (2004) investigate a client-server approach for visualizing complex 3D models on thin clients applying real-time MPEG-4 streaming to compress, transmit, and visualize rendered image sequences. They identify the MPEG-4 encoding speed as bottleneck of client-server 3D rendering, and devise a fast motion estimation process for the MPEG-4 encoding.

2.2 Data Model

The main purpose of navigation application is to interpret the process such as “whereby people determine where they are, where everything else is, and how to get to particular objection or places” (Jul and Furnas 1997). The task can be distinguished into three kinds, naive search, targeted search, and exploration (Darken and Sibert 1996). To do this, users need builds up a mental model of the virtual environment by forming linear maps and combining them to spatial maps (Ingram and Benford 1995), and corporate task-based constraints on the navigation parameters (e.g. viewer position and orientation).

At the present time, a lot of work has been done which mainly aim at how to enhance the visualization efficiency and many sophisticated data structures have been designed. For example, a number of LOD algorithms have been developed to create a hierarchy of mesh refinement operations to adapt the surface and decimate polygons thus reducing complexity of computation without affecting the quality of scenes. (Lindstrom et al. 1996) introduce a real-time smooth and continuous LOD reduction using a mesh defined by right triangles recursively subdivided according a user-specified image quality metric. Some hierarchies use Delaunay triangulations (e.g. Cohen-Or and Levanoni 1996; Cignoni et al 1997; Rabinovich and Gotsman 1997) while others allow arbitrary connectivities (e.g. De Floriani et al 1997; Hugues Hoppe 1998; El-Sana and Varshney 1999). In (Duchaineau et al. 1997), the authors introduced ROMAing method as a very efficient algorithm based on triangle diamonds managed with split and merge operations performed using priority queues. The algorithm now is widely used in games industry, but its implementation is tedious according to (Blow 2000). In 2002, (Levenberg) propose to reduce the CPU overhead of the previous binary-triangle-tree-based LOD algorithms by manipulating aggregate triangles instead of simple triangles.

But applications of 3D navigation suffer from a lack of data standards and flexible distribution techniques. The virtual 3D models frequently are implemented as graphic models without explicitly modeling semantic and topological relations. Therefore, the data can only be used for visualization purpose but not as a basis for higher-level functionality such as simulations, analysis tasks, or spatial data mining.

3. OFFLINE DATA PREPARATIONS

Before geo resources can be efficiently accessed at runtime, the datasets including digital elevation models (DEM), aerial photographs, entity models and their facade images need to be prepared and organized in an offline process. The main purpose of this process is to define the data structures, compress the spatial data, reduce the data redundancy and enhance the rendering efficiency.

In recent years, there have been many techniques purposed to partition and organize data with multiple resolution into hierarchical structure. The most common ones are Quad-tree, BSP (Binary Space Partitioning) tree and Octree. In our approach, we designed a pyramid mode for multi-resolution virtual environment and partition the whole world into different levels and block in terms of latitude and longitude. As each level of pyramid data has its own specific storage unit (Here these units are called as tiles) and access needs, for each level *l*, with grid spacing $S_l = S \times 2^{-l}$ in world space, it is let the desired active region be the square of size $nS_l \times nS_l$. Here the parameter of S represents the total space of the area.

When multi-resolution pyramid is generating, each level of pyramid is represented by hierarchical quad-tree data structure and one tile corresponds to a certain range of region, where the width and height of the tiles are measured in decimal degrees. Child nodes are generated from a parent node by equally splitting the parent node tile into 4 quadrants. Each child nodes tile has half the width and height of the parent node tile. The top-level node in this tree structure represents the area of the entire tile, its children each represent one fourth of the terrain area, their children in turn each cover one sixteenth of the area (see figure 1). The root node of the tree is denoted as levels of 0, and is centered on the latitude $\phi = 0^\circ$ and longitude $\lambda = 0^\circ$, so

the root level of tile has the width and height spans the whole globe world. By this means, the area of any node can be specified by its east (*E*) and west (*W*) longitude and north (*N*) and south (*S*) latitude. For the root node, $E_0 = 180^\circ$ $W_0 = -180^\circ$ and, $N_0 = -90^\circ$, $S_0 = 90^\circ$.

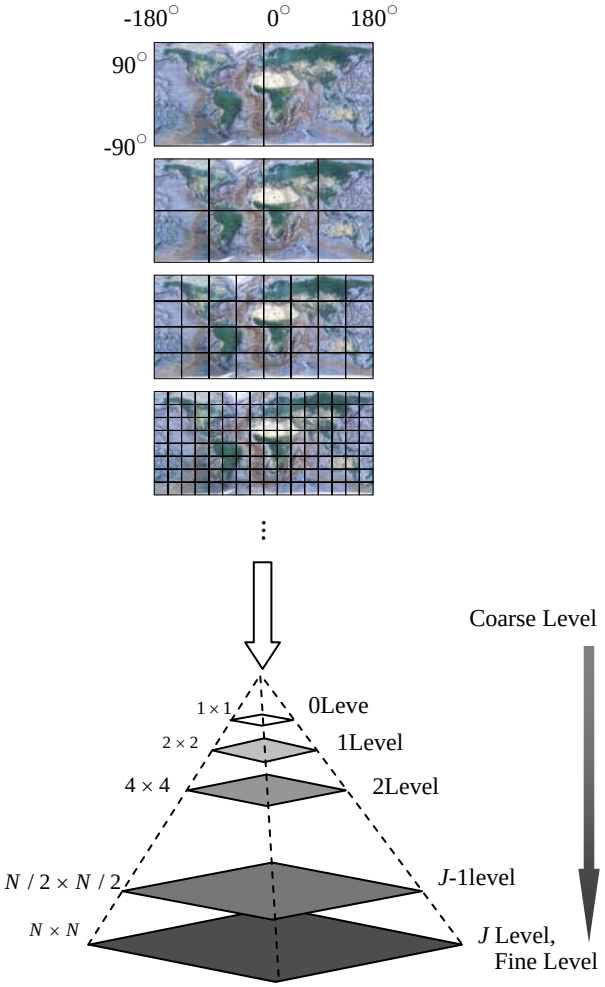


Figure1. Multi-Resolution Pyramid model

The texture tiles are organized in quad-tree structure too (see figure2.), which each texture is linked to a unique node in the tree and each node is thus associated with a coverage area, or a tile. Usually, real-time rendering of massively textured 3D scenes involves two major problems: Large numbers of texture switches are a well-known performance bottleneck and the set of simultaneously visible textures is limited by the graphics memory. So it need use different resolution texture to real-time render the massively textured scenes. The basic principle of multi-resolution texture rendering can be described in: In each frame, the texture resolution is chosen in the way that the textel-per-pxiel ratio is always near to 1 so that the amount of necessary texture data remains small (Henrik Buchholz, J. Döllner, 2005).

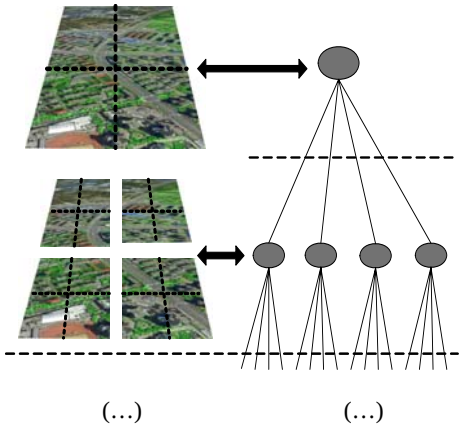


Figure2. Structure of the texture quad-tree. Each node represents a certain scene part.

For generating pyramid, each level of pyramid is a single storing unit, e.g. a file, and each node of the tree stores a single texture image and each texture image of level *n* is decomposed into list of samples containing the colour of R (red), G (green),B (blue), and the sample's location given as latitude ϕ and longitude λ. Depending on the location, each list contains the sample is assigned to a list such that each list contains the samples of the covered area of one tile at the given resolution level *n*. In addition, each node stores a distance variable which represents the minimum distance between the view position and the node's bounding box to ensure that the node's texture resolution is sufficiently high. The scene geometry is stored in the leaf node. Each leaf node contains the triangles of its corresponding scene part and the related subset of the original texture of the input scene.

In this paper, we introduce mipmap texture into virtual environment, just to make it be one part of the pyramid mode and choose the appropriate mipmap-level for the corresponding area in texture space at runtime. The basic principle of mipmap is: the nearer to the viewpoint, the higher resolution texture is required, that is when the projected scale of the surface increases, interpolation between the original samples of the source image is necessary; as the scale is reduced, approximation of multiple samples in the source is required. To reduce the computation implied by these requirements, a set of prefiltered sourced textures may be created and a succession of levels which vary the resolution from the original data is represented (see figure3.).

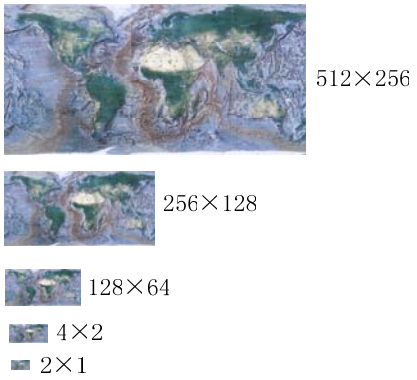


Figure3. Various resolution textures of Mip map

4. 3D SCENES RENDERING

For a given location, the server gets the surrounding DEM, models data and other additional information from web map data server. In our approach, only those tiles which lie in frustum view region will be loaded, and search those tiles are by grid index (see figure 4a). If viewpoint moves over an adjacent tile the algorithm will tend to maintain a square of tiles centered on this new tile (which will be load in client memory new and becomes the current tiles). At the same time, the algorithm will remove some far tiles which are not within the field-of-view in order to free memory for the fetching of new tiles (see figure 4b). As figure 3a indicated, most of the memory of mobile device was consumed at the step. Note that the algorithm implicitly handles the case where viewpoint jumps to a new tile that is not adjacent to the current one. The quad-tree representation of tile data enables very fast view frustum culling.

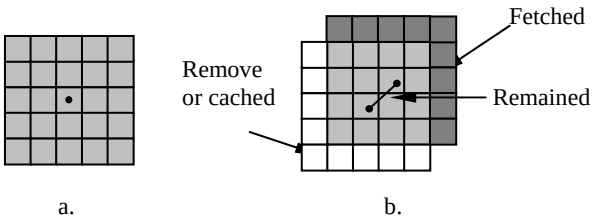


Figure4. Tiles management and adaptive loading
a). A square area centered on the viewpoint.
b).Square area preservation on viewpoint move

In order to enhance rendering efficiency, numerous methods for mesh simplification have been developed on the last decade. In this paper, the simplifying approach is based on the terrain height field and terrains are represented in various resolution meshes. Firstly, the full terrain height-field is divided into regular tiles, and then the appropriate level of detail is computed and generated dynamically, allowing for smooth changes of resolution across area of the surface (see figure 5), those even areas are represented in low resolution meshes and the uneven areas are represented in high resolution meshes.

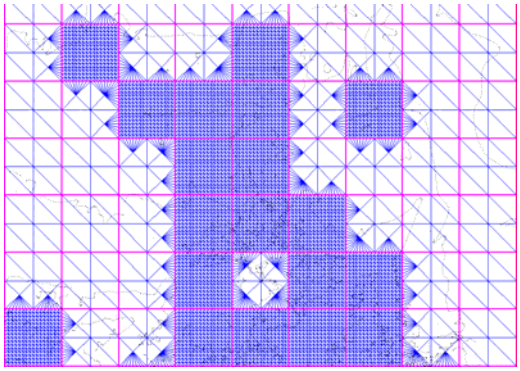


Fig5. Simplified mesh based on terrain height fields

5. CLIENT-SERVER ARCHITECTURE FOR 3D NAVIGATION

There are stand alone 3D applications which can now run on mobile devices. However, for the limited capabilities of the mobile devices and aiming to provide the users a panorama scene that fits with the real surroundings navigation, our approach is implemented based on client-server architecture (see figure 6).

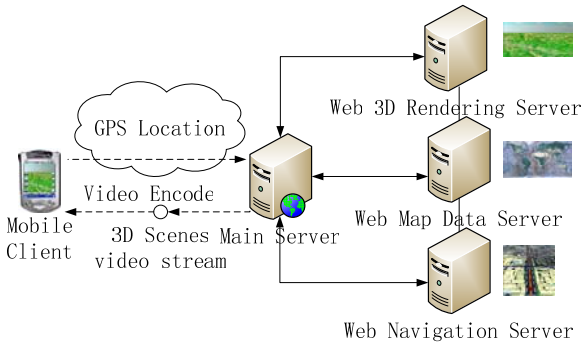


Figure6. Client-server architecture of 3D mobile navigation system

The server system provides a web service interface to the virtual environment, and it is responsible for handling requests sent from the clients. The 3D models' datasets are hosted on a web map data server, and the server interprets and controls navigation commands, then sends the requests to a web 3D rendering server to generate a virtual panorama by rendering the DEM and a web navigation server generates 3D scenes using the panorama images and some other nodes corresponding to meta information. The rendering component encodes the frames into video stream as 3D scene files. The clients can communicate with the server by exchanging SOAP messages and the 3D scene files are finally sent to a mobile client to show.

A mobile device as a thin client system need not contain any application of navigation software, it only needs capabilities for receiving and playing the multimedia, capturing the user requests, sending and receiving SOAP message.

6. EXPERMENT AND CONCLUSIONS

Due to the complexity of the real world scenario and the vast computational power required to achieve a usable performance, navigation in 3D environment is still a tough task. In our experiment, we handle complex 3D scene models by culling areas outside of the field of view, and by using multi-resolution models to reduce the data.

6.1 Experiment and results

This paper proposed an efficient framework for resource management and texture handing regarding online and offline procession. The Pc server's primary configurations are shown in table 1, and the client platform we selected was a WindowCE device as for its multimedia capability and easy of programming with Embedded C++.

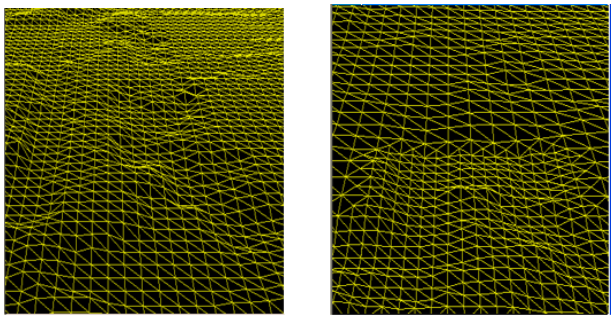
CPU	Pentium(R) 5, 2.66GHz
GPU	ATI MOBILTY RANEON, 7500
Memory	1.00GB
Storage	250GB
Operating System	WinXp, Professional, SP2

Table 1. Configurations of Pc Server

In the experiment to test the efficiency of network transmission, we found the major bottleneck is the massive 3D models'

datasets transfers, especially with slow home or mobile modems. Level of details improves the rendering speed but does not affect the download time, and on the other hand, the raw 3D scene model data imply severe drawback for data security and copyright issues. So we transmit only video sequences but no raw data to the mobile clients. After the 3D scenes data transferred to the mobile client, the frame rate was sufficient for conducting the navigation test on mobile client.

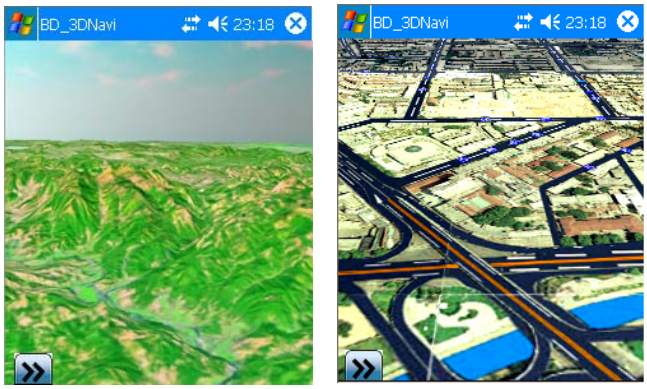
In the experiment which measured the difference between simplify scene and primal scene, we recorded the amount of primary triangles transmit to client memory is about 70,000 (see figure 7a), and after being simplified, the amount of triangles reduced to 16064, almost 77.1 percent triangles are removed or cached (seeing figure 7b).



a b

Figue7. Comparison between primary scenes and simplified scenes. a) Primary terrain scenes. b) Simplified terrain scenes

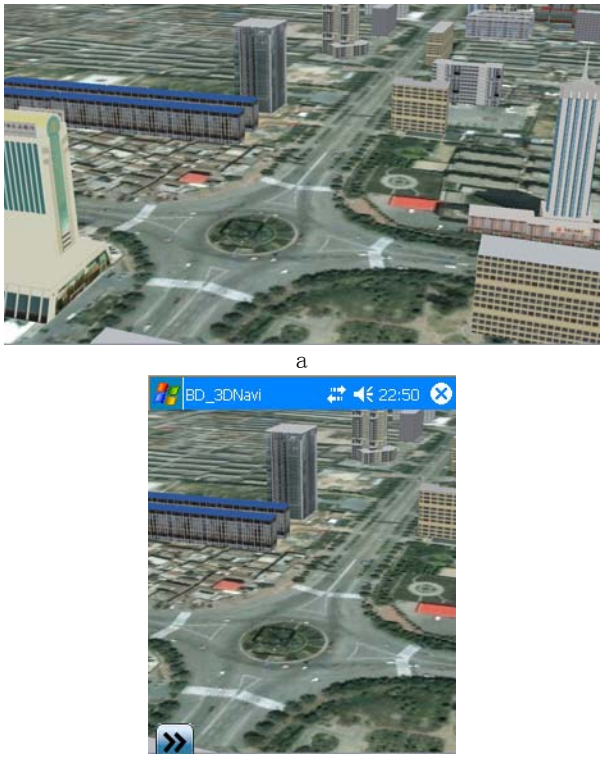
In the experiment which test the efficiency of 3D scenes generating and rendering, the test area we selected spans about 200Km×200Km, and it was covered by 30.0m resolution ETM color photography and some attention areas were covered by 1.0m resolution color aerial photography (about 3.5 GB), as well as 120 MB of DEMs which the highest resolution is 16m spacing and the lowest resolution is 256m spacing. On Pc server, we recorded the rendering efficiency about 80-120 fps with 50%-60% CPU utilization and a delay inserted between the frames to maintain constant frame rates.



a. b.

Figure8. Panorama view of large terrain view with 30.0m resolution ETM color image (a) and city view with 1.0m resolution color aerial image (b) on mobile client

In the city model experiment, over five hundred building models were rendered, and each building contains hundreds of triangles and the facade texture of one building is about several million bytes. The average frame rate on servers was about 30-50 fps for the original texture, and the textures were DDS compressed and using mipmap textures, the average frame speed up to 70-100fps during real-time rendering.



a

b.

Figue9. Panorama of the city models. a) City models on Pc servers. b) Sketch the city models on mobile client

6.2 Conclusions

Mobile devices currently have the capability to request and display 3D panorama scene. This paper proposed an approach to generate 3D environment visualization system for personal navigational purposes that handles large heterogeneous datasets at multi-resolutions. The attached GPS provides the location information, and network servers provide the data and visualization processing. In the stage of offline data preparations, the full area is divided into regular tiles and a pyramid mode for multi-resolution virtual environment is generated. Having the virtual environment being divided into zones helps the users to narrow down their search towards or within the intended zone or category only, and it only need transmit compressed imagery that is actually requested by the users. By the client/server mode, the presented approach allows mobile applications to provide users interactive access to complex 3D scene models including high-resolution 3D terrain geometry, 3D building geometry, and textures. In particular, the server can be optimized for processing large-scale 3D scene models using high-end computer graphic hardware, whereas on mobile clients, there only multimedia capabilities are required.

6.3 Future work

The major problem with detailed 3D scene models is the big size, which affects both the rendering speed and the download

time. Especially in city, the models of cities and their details can be nearly infinite. In future we need to explore other image coding algorithms and graphics optimization techniques such as occlusion culling.

7. REFERENCES

J ü rgen Döllner, Benjamin Hagedorn, Steffen Schmidt, An Approach towards Semantic-Based Navigation in 3D City Models on Mobile Devices, 3rd Symposium on LBS & TeleCartography, pp. 171-176, November 2005.

Martin Hachet, Joachim Pouderoux, Sebstian Knödel, Pascal Guitton, 3D Panorama Service on Mobile Device for Hiking. 2007.

Nazrita Ibrahim, Nurul Fazmidar Mohd Noor, NAVIGATION TECHNIGUE IN 3D INFORMATION VISUALISATION, IEEE, 2004.

Antti Nurminen, A Platform for Mobile 3D Map Navigation Development, MobileHCI'06, September pp. 12-15, 2006, Helsinki, Finland.

Peter L. Guth, Pocket Panorama: 3D GIS on a Handheld device, IEEE, 2004, the Fourth International Conference on Web Information Systems Engineering Workshops.

Ismo Rakkolainen, Teija Vainio, A 3D City Info for mobile users, Computers & Graphics 25 (2001) pp. 619-625.

Luca Chittaro, Stefano Burigat, Location-aware visualization of a 3D world to select tourist information on a mobile device.

Kevin Peterson, Three Dimensional Navigation, IEEE, 2002.

Henrik Buchholz, Jürgen Döllner, View-Dependent Rendering of Multiresolution Texture-Atlases, IEEE Visualization 2005 October 23-28, Minneapolis, MN, USA.

Michael Brünig, Aaron Lee, Tuolin Chen, and Hauke Schmidt, Vehicle Navigation Using 3D Visualization, IEEE, 2003.

Mandun Zhang, Linna Ma, Xiangyong Zeng, Yangsheng Wang, Imaged-Based 3D Face Modeling, IEEE, 2004, The International Conference on Computer Graphics, Imaging and Visualization (CGIV'04).

Miran Mosmondor, Hrvoje Komericki, Igor S. Pandzic., 2006. 3D Visualization on mobile devices. Telecommun Syst,32, pp. 181-191.

Joachim Pouderoux, Jean-Eudes Marvie, Adaptive Streaming and Rendering of Large Terrains using Strip Masks, Proceedings of ACM GRAPHITE 2005 - 2005

GRAPH-BASED URBAN OBJECT MODEL PROCESSING

Kerstin Falkowski and Jürgen Ebert

Institute for Software Technology
University of Koblenz-Landau
Universitätstr. 1, 56070 Koblenz, Germany
{falke|ebert}@uni-koblenz.de
http://www.uni-koblenz-landau.de/koblenz/fb4/institute/IST/AGEbert

KEY WORDS: Urban, Model, Metadata, Data Structures, Algorithms, Processing, Services.

ABSTRACT:

Urban object models are valuable assets that allow reuse in different applications. Besides the need for exchange formats there is also the need for comprehensive, efficiently processable data structures for such models. This paper presents a graph-based schema for integrated models of urban data, that is an adaption of the comprehensive CityGML approach. It defines an explicit graph representation and thus is well-suited to efficient processing algorithms. The paper demonstrates how appropriate light-weight components realizing different kinds of services on models can be used for consistently processing semantics, geometry, topology and/or appearance of graph-based models compliant to that schema. Several examples are given.

1 INTRODUCTION

Urban models are valuable assets that should be constructed once while being used multiple times in different applications. Therefore the exchange of 3d city models between different tools is indispensable. Various XML formats are being used to achieve interoperability between tools. These formats (e.g. CityGML (Groeger et al., 2008)) are able to carry topological, geometric, semantic, and appearance information, but in different forms and to varying extent.

Applications, like tools for the automatic extraction of topographic objects, build on these urban object models and improve, transform, and analyze them in different ways. XML-technology (e.g. XSLT and XQuery) is widely used to support these activities, but this technology is not well-suited for the implementation of the various algorithms on urban objects which come from the areas of algorithmic geometry, computer graphics, and image recognition, since the necessities of efficient content-based traversal of all relevant information is only hard to realize in the essentially tree-like structures supplied by XML.

Therefore, a comprehensive, efficiently processable data structure for urban objects is essential. Geographic information systems share this necessity with route guidance systems, where a graph-like internal representation of data is used for the computation of routing information.

In this paper, we present an approach for the efficient storage, analysis, and manipulation of city models using graphs and for the development of application specific components collectively working on an integrated, efficient graph representation of city models. Import/export from/to CityGML is tackled, as well. [1]

After a short overview of the state of the art in section 1.1, section 2 shortly introduces the employed graph and component concepts. Section 3 describes the graph-based integrated model schema with all its aspects, and section 4 shows how quite different kinds of functionalities can be implemented on such a model by independent components. Section 5 concludes the paper.

[1] The project is funded by the DFG (EB 119/3-1).

1.1 State of the art

There are several XML-based modeling languages for urban objects. The *City Geography Markup Language (CityGML)*[2] is a common information model for the representation of 3d urban objects and an official standard of the Open Geospatial Consortium (OGC) since August 2008 (Groeger et al., 2008). Besides representing geometry, CityGML can also be used to model topological and semantic properties of 3d city models and to attach appearance information like textures.

Models described using CityGML can be rendered by *Ifc-Explorer for CityGML*[3] from the Institute for Applied Computer Science, Forschungszentrum Karlsruhe or the *LandXplorer CityGML Viewer*[4] from Autodesk and by *Aristoteles*[5] from the Institute for Cartography and Geoinformation, University of Bonn.

Besides CityGML there are other languages for the representation of 3d urban objects. One common approach is the OGC standard *Keyhole Markup Language (KML)*[6]. CityGML uses a subset of the OGC standard *Geography Markup Language (GML)* (Cox et al., 2001) for geometry representation, KML derived his geometric elements from GML. KML is often combined with the COLLADA[7] exchange format for 3d assets. Another 3d modeling language is *Extensible 3D (X3D)*[8], the successor of the *Virtual Reality Modeling Language (VRML)* standard.

2 BASIC TECHNOLOGIES

2.1 TGraph technology

For the efficient manipulation of urban object models with all their aspects a versatile and powerful basic technology is needed. In the context of this work TGraph technology is used.

[2] http://www.citygml.org, http://www.citygmlwiki.org
[3] http://www.iai.fzk.de/www-extern/index.php?id=1570
[4] http://www.3dgeo.de/citygml.aspx
[5] http://www.ikg.uni-bonn.de/aristoteles
[6] http://www.opengeospatial.org/standards/kml
[7] http://www.khronos.org/collada
[8] http://www.web3d.org/x3d

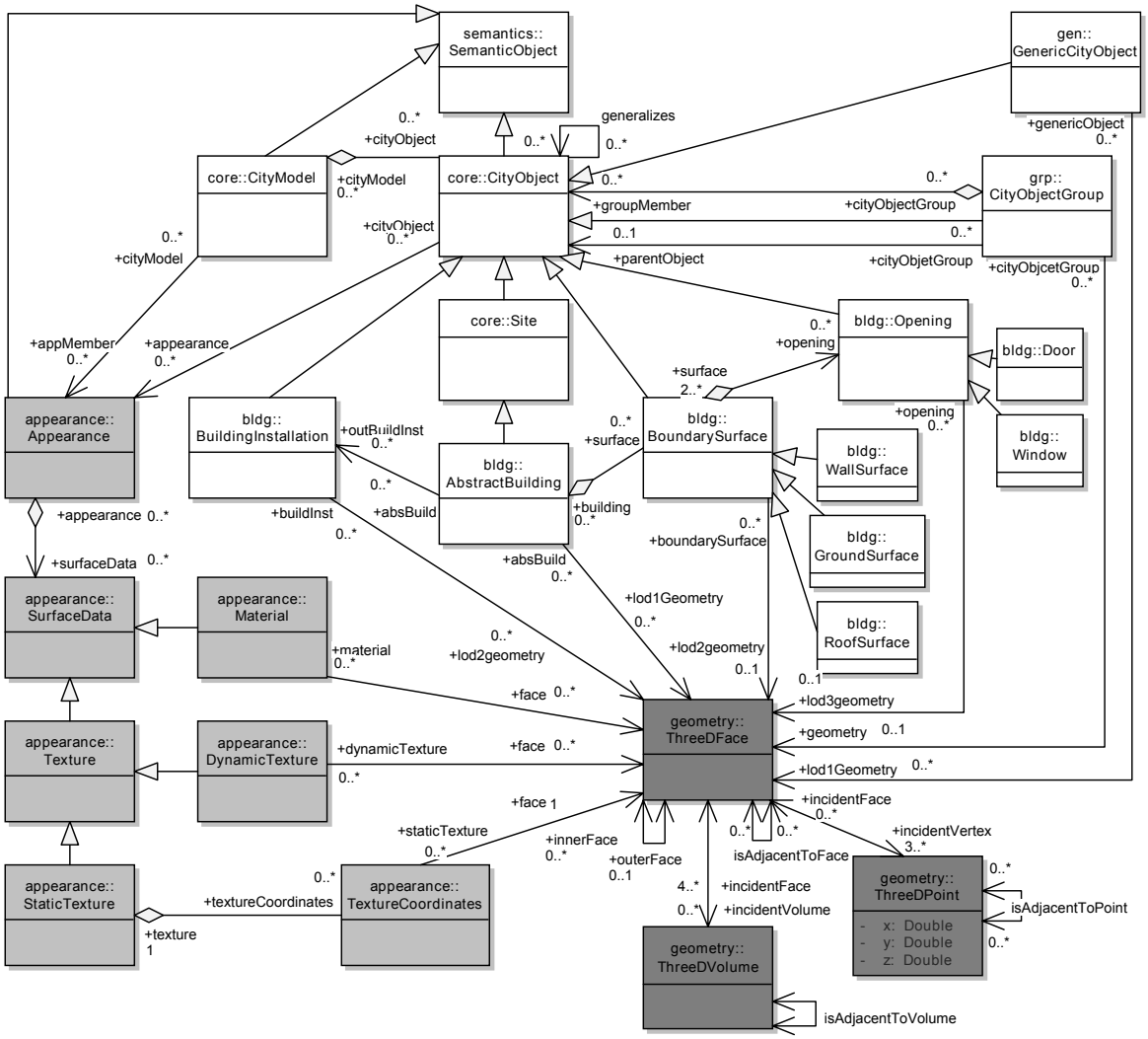


Figure 1: The integrated model schema as a grUML diagram. *Semantic entities* are colored white with namespace "sem", *appearance entities* are colored gray with namespace "app" and *geometry/topology entities* are colored dark gray with namespace "geo/top".

TGraphs are directed graphs whose vertices and edges are typed, ordered and attributed. Their structure, types and attributes help to model the different aspects (topology, geometry, semantics, and appearance annotation) of urban objects in a common integrated data structure. TGraphs are supported by a powerful API (JGraLab[9]) in combination with a *graph query language* (GReQL) and a corresponding UML-based *metamodeling approach* (grUML). grUML is a subset of UML class diagrams which allows the specification of classes of TGraphs on the schema level (Ebert et al., 2008). Figure 1 contains an example.

2.2 Lightweight component model

If all relevant data of the urban object model are stored in a TGraph, all processing of the model can be encapsulated in appropriate components working on this particular TGraph.

The work described here is based on a *light-weight Java component model* which is employed for the different processing activities on the model (see section 4). The component concept is basically an extension of the well-known *strategy pattern* (Gamma et al., 1995). Every component has a *definition* in the form of a Java interface which describes its service and at least one *implementation* in the form of a Java class.

[9] `http://jgralab.uni-koblenz.de`

Components are serializable and get their data to process as *arguments* of their `execute()`-methods. Further data that influence their work are handled as *parameters* which have a default value and are manipulated via *getters* and *setters*. For example some processing steps can be configured by parameters (like thresholds).

3 THE INTEGRATED MODEL SCHEMA

The internal representation of urban object models by TGraphs has to be specified by a metamodel, called *schema* in the following. This schema defines the set of compliant TGraphs. Classes define the possible vertex types, and associations define the edge types. The attributes of vertices and edges can be added according to the well-known UML notation, as well. Edge direction is visualized by arrow heads, though it should be noted, that TGraph edges are traversable in both directions by algorithms.

Figure 1 shows the main parts of an integrated schema which defines a set of TGraphs for urban objects. (To improve readability all enumeration types and some semantic subclasses as well as attributes are elided.) This schema is inspired by and partially derived from the CityGML 1.0 schema. Especially it follows the idea to separate the four relevant aspects of an urban object model (namely topology, geometry, semantics, and appearance).

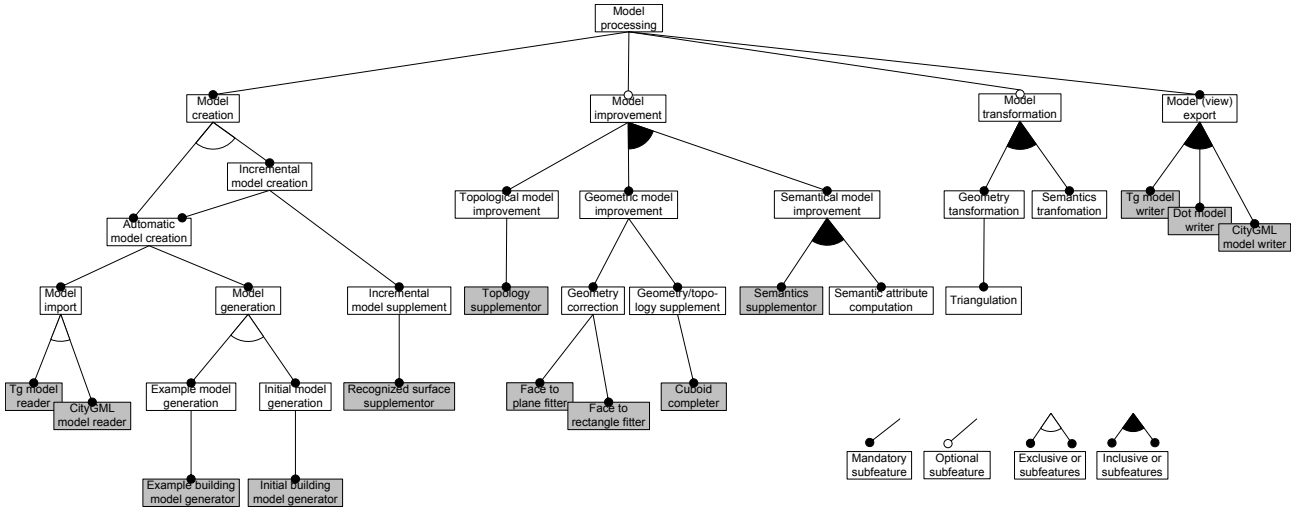


Figure 2: The main components as feature diagram. *Component groups* are colored white, *concrete components* are colored gray.

The schema contains *semantic entities*, *appearance entities* and *geometry/topology entities*. The semantic part contains entities from the subpackages Core (namespace "core"), CityObjectGroup (namespace "grp"), Generics (namespace "gen") and Building (namespace "bldg"). It should be noted that in principle also other ontologies might be used for the semantic part.

This grUML schema extends the tree-like XML schema of CityGML to a real graph-based schema, that (meta-)models the entities and relations of urban objects much more explicitly.

In CityGML models multiple occurences of the same object can mostly be modeled by defining the object once and referencing it using XLink[10]. But this is not possible for every object. As an example, the GML specification offers the definition of the control points of a `LinearRing` (exterior of a `Surface`) using the types `DirectPosition` or `PointProperty`. The first is used, if the control points are used only in this geometry element, the second is used used, if the control points may be referenced from other geometry elements. CityGML restricts these possibilities to `DirectPosition`. This means in CityGML models for every occurence of the same real world point as control point of the surfaces of a building there is a new `DirectPosition`. And even if XLinks are used, they often can not be processed sequentially and their interpretation is time-consuming.

Using the integrated model schema of figure 1, every entity in an urban object model exists only once as a node and all its uses and occurences are modeled by edges. This explicit, strongly linked representation reduces redundant information and enables automatic model processing by a very large class of algorithms. It is easy to import and export CityGML models using LODs 1-3 to and from an integrated model.

3.1 Geometry/topology schema part

The geometry/topology part of the integrated model schema differs from the CityGML schema. CityGML uses a subset of GML to represent geometric entities as a *boundary representation* (Foley et al., 1990, Herring, 2001). But the geometry/topology part of the integrated model schema is not based on this GML subset, since entities and relations are not represented explicitly enough and the same geometric objects may appear more than once in the same model.

[10] http://www.w3.org/TR/xlink

Here, geometry and topology are modeled as another kind of boundary representation, namely as an extended *vertex-edge-face-graph* (*v-e-f-graph*) similar to the well-known and highly efficient *Doubly Connected Edge List (DCEL)* representation (Muller and Preparata, 1978). A geometric object consists of *3d points*, *3d faces* and *3d volumes*, modeled as typed nodes connected via edges. The *geometric information* is encoded in the *attributes of the 3d points*, and the *topological information* is represented by the *edges between the geometric entities*.

3.2 Semantics schema part

The semantics part of the integrated model schema is based on the CityGML modules *Core*, *CityObjectGroup*, *Generics* and *Building*. Thus, terms like "building", "wall surface" and so forth can be used without further explanation in the following. Each of the mentioned modules is packed in its own subpackage.

3.3 Appearance schema part

The appearance part of the integrated model schema is oriented at the CityGML appearance module. But the different kinds of surface data (material, different kinds of textures) are directly related to the 3d faces they shall be applied to. The model allows *static and dynamic textures*, but dynamic textures are preferred. A dynamic texture consists of an image and a transformation matrix containing values to compute 2d texture coordinates for existing 3d points concerning the given image. By using dynamic textures, texture coordinates can be updated during model export if their corresponding 3d points have changed during model improvement.

4 INTEGRATED MODEL PROCESSING

The integrated model schema defines the class of TGraphs that represent urban object models with all their aspects. There are a lot of possible *processing activities* for integrated models, which are introduced in the following. Figure 2 gives an overview over such processing activities and their dependencies in the form of a *feature diagram* (Czarnecki and Eisenecker, 2000).

Here, the components constitute a product line (Pohl et al., 2005) where *features* are implemented by *Java components* (subsection 2.2). (The components are referenced by identifiers written in `typewriter` style.)

This chapter presents some of these processing components in more detail in order to prove on an example basis that all kinds of processing is possible on integrated models based on TGraphs.

The (intermediate) results of the different processing activities are exported using the `CityGMLModelWriter` component (subsection 4.5) and the XML text is rendered using the IfcExplorer for CityGML (section 1.1). IfcExplorer encodes different (semantic) parts of CityGML models using various colors. Wall surfaces are rendered gray, ground surfaces dark gray, roof surfaces red, doors dark blue, windows light blue and nearly transparent and all other faces cyan. (Unfortunately this distinction is hardly visible in the black-and-white versions of this article.)

4.1 Example

The functionality of the components is demonstrated on the basis of a model of one simple example building, which may be created using the `ExampleBuildingModelGenerator` (subsection 4.2). The full model consists of one ground surface, four wall surfaces, four roof surfaces, one door and five windows, its geometry contains fifteen 3d faces and thirty four 3d points. The user can choose, which model parts should be generated and how they should be connected. Figure 3 shows the full model. Since semantics, geometry and topology of this example model are well known, it is used as example model for most of the components mentioned in the following.

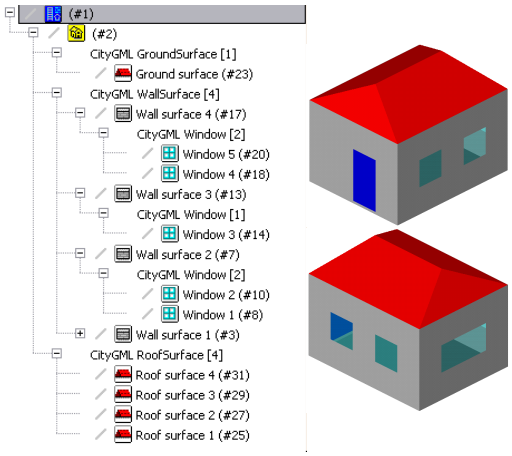


Figure 3: Full example model.

4.2 Model creation.

First of all, an integrated model has to be created. There are two kinds of automatic model creation, namely *model import* and *model generation*.

Model import. During model import an existing model is read from a file. `TgModelReader` reads an existing integrated model from a .tg-file (the JGraLab file format) and `CityGMLModelReader` reads an existing CityGML model from an .xml-file and transforms it into an integrated model. This import respects the CityGML[Appearance,Building,CityObjectGroup,Generics][11] profile. Semantic objects from other CityGML modules are ignored at present.

Model generation. During model generation an integrated model is created from scratch by a list of creation steps which are hard-coded in Java. The component `ExampleBuildingModelGenerator` constructs the complete example model from figure 3 that contains all four integrated model parts. `InitialBuildingModelGenerator` constructs incomplete models which function as bases for incremental model supplementation activities, which are not explained in further detail here.

[11] The Core module is not mentioned in CityGML profile names, because it belongs to every profile

4.3 Model improvement

The advantage of a graph-based representation of 3d models becomes clear if elaborate algorithmic activities are applied to them. Such activities are especially needed if the imported model is still unprecise and incomplete, for instance because it consists of raw data delivered by some object extraction tool (Falkowski et al., 2009).

Then, the raw models might have to be improved algorithmically. This includes *topological*, *geometric* and *semantic model improvement*. Geometric model improvement may even be specialized into *geometry correction* and *geometry/topology supplement*.

Topological model improvement. During topological model improvement different kinds of topological information are added to an integrated model. The component `TopologySupplementor` complements an integrated model by adding implicit topological dependencies as explicit arcs in the graph. It may connect all neighboring faces of a 3d face by `isAdjacentTo`-edges and all neighboring 3d points accordingly to a 3d point, if they are not related yet. Furthermore it may add all 3d faces that lie in another 3d face as inner faces. The component uses the geometric/topologic model part, but changes only topology.

Topological model improvement should always be the first improvement step, since most of the later processing steps are based on computational geometry algorithms that assume complete topological information.

Geometry correction. In raw models, the computed 3d coordinates are often only known approximately. This may lead to (slightly) distorted models. See figure 4 as an example.

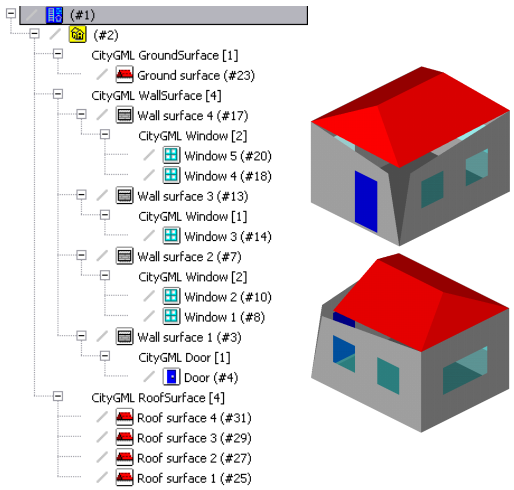


Figure 4: Geometry correction: Example model with "wrong" 3d point and therefore with 4 non-planar faces.

During geometry correction the geometry information of an integrated model (i.e. the x-, y- and z-coordinates of 3d points) is corrected. The `FaceToPlaneFitter` tests the planarity of all 3d faces and makes them planar, if they are not. The `FaceToRectangleFitter` tests the squareness of all 3d faces and makes

them rectangular, if they are nearly squared. Both components use appropriate approximation algorithms and both use the geometric/topologic model part, but change only the geometry. This correction transfers the model of figure 4 to the one in figure 3.

Geometry/topology supplement. Models extracted from 2d images are usually incomplete, since hidden information is missing. For urban data (sometimes) plausible assumptions may be made about the 3d-structure of the objects (e.g. they may be assumed to be cuboids).

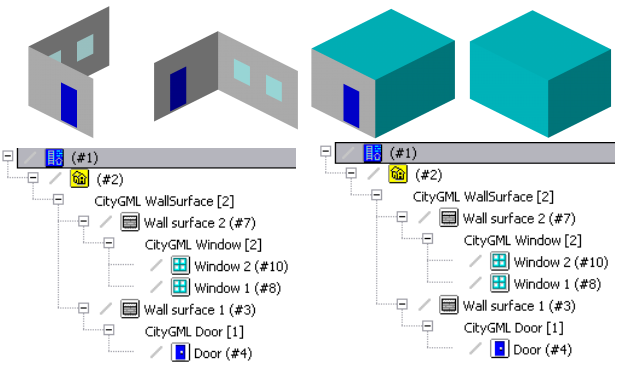


Figure 5: Geometry/topology supplement: Incomplete Example model (left), supplemented example model (right).

During geometry/topology supplement different kinds of geometric and topological information are added to an integrated model. The `CuboidCompleter` tests if there are incomplete cuboids in the integrated model and completes them by adding mirrored inverted copies of existing 3d faces (figure 5). The component uses the geometric/topologic model part, and enhances geometry as well as topology.

Semantic model improvement. Given a corrected and supplemented model, also semantic information might be inferable and should be added to the model.

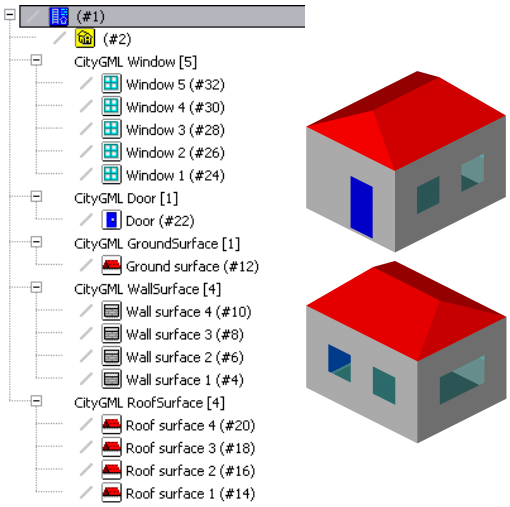


Figure 6: SemanticsSupplement: Model without relations between building and boundary surfaces as well as boundary surfaces and openings.

During semantic model improvement different kinds of semantic information are added to an integrated model. The component `SemanticsSupplementor` complements an integrated model by adding implicit semantic dependencies as explicit relations. It acts on the assumption, that if an object belongs to an aggregation, its parts also have to belong to this aggregation as well and vice versa. The component adds openings or building to a city model, if their related boundary surfaces belong to this city model. Figure 6 shows a an example of a semantically poor model which is be transformed into the full model of figure 3 by this component. The component uses the semantics and the geometric/topological model part, but changes only semantics.

4.4 Model transformation

A general class of processing activities is the modification of an integrated model by some kind of model transformation. There are *geometry/topology transformations* and *semantic transformations*. An example for geometry/topology transformation could be *triangulation*. An example for latter might be *changing the CityGML like semantics part* into one according to a proprietary ontology.

4.5 Model export

In general the integrated model or at least parts of it have to be stored persistently after processing. During model export an integrated model is written to a file.

The component `TgModelWriter` writes a full integrated model to a .tg-file. If the exported .tg-model is imported again, no information will be lost.

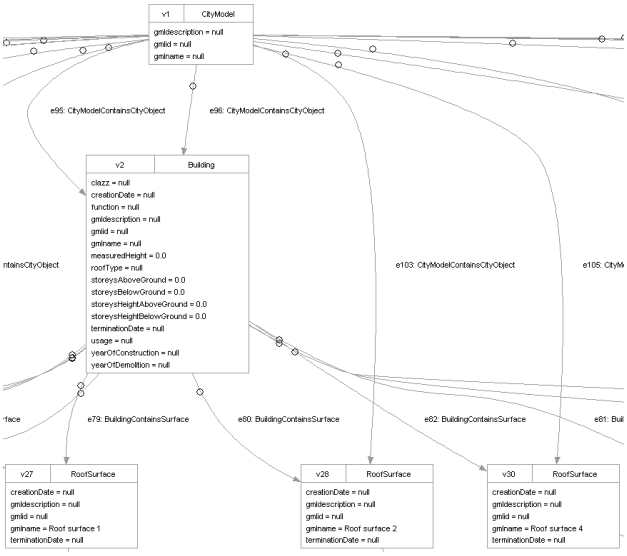


Figure 7: Extraction of the example graph, exported to .dot format and rendered via dotty, a Graphviz tool.

The component `DotModelWriter` writes an integrated model to a .dot-file, the standard file-format of the *Graphviz*[12] graph visualization software (figure 7). The result can be processed further using Graphviz.

The `CityGMLModelWriter` writes the integrated model via a special graph traversal algorithm as a CityGML[Appearance,Building,CityObjectGroup,Generics] model into an .xml-file. The user can influence the result by choosing the LOD and the kinds of textures to be written. The result can be processed further by other tools. For example, it may be rendered via any appropriate CityGML Viewer (see section 1.1). If the exported .xml-model is imported again, information might be lost, since the integrated model contains more information than those covered by CityGML.

[12] `http://www.graphviz.org`

4.6 Model analysis

Many more activities might be implemented on the integrated model, since all kinds of queries may be posed on it. Thus, a further processing activity is the analysis of the integrated model. This activity is not mentioned in the feature diagram in figure 2, since it is carried out as part of nearly every other integrated model processing activity. All tests concerning existing model elements and/or their properties or relationships are *model analysis steps*. There can also be *transversal analyses*, regardind coherences of a whole model, a part of a model (e.g. one building) or a special view to a model (e.g. geometry). Using the TGraph structure traversal and analysis of the integrated model is done repeatedly during runtime using the graph API and/or GReQL queries. Transversal analyses are particularly supported by the graph API. For example it offers iterators for all nodes or edges of a special type (and it subtypes) in the whole model.

Listing 1: Building analyser results.

```
Building 1

Id: 2
Name: Example building
Description: Example building model for testing.

Year of construction: not known
Year of demolition: not known

Number of appearances: 0
Number of building installations: 0
Number of building parts: 0

Number of boundary surfaces: 9
Number of wall surfaces: 4
Number of roof surfaces: 4
Number of ground surfaces: 1
Number of openings: 6
Number of doors: 1
Number of windows: 5

Number of 3d faces: 15
Number of 3d points: 34

Lowest 3d point: Point 1: (0.0, 0.0, 0.0)
Highest 3d point: Point 9: (2.0, 1.0, 4.0)
Height: 4
Width: 4
Depth: 5
Volume: 68
```

To demonstrate the usage of querying with GReQL an additional component `BuildingAnalyser` was developed, that writes information about all buildings of the integrated model into a .txt-file. The file contains different kinds of information. At first there is *semantic attribute information* like name, description and year of construction/demolition of the building. Moreover there is *semantic entity information* like the number of wall, ground and roof surfaces, the number of doors and windows, and so on. Furthermore there are *geometric information* like the count of points and faces of the building geometry or the lowest and highest point of a building. And there is *inferred semantic information* computed using semantic background knowledge in combination with geometry information, like the building height, the building volume, and so forth (listing 1).

5 CONCLUSIONS AND FUTURE WORK

This paper showed how geometric, topological, semantic and appearance information can be integrated in one integrated graph model. The class of models was defined by an *integrated model schema*. Graph representation gives rise to all kinds of algorithmic processing, some examples of which were given, including model creation, improvement, transformation, analysis and export. Using a *lightweight Java component model* some example components were implemented and illustrated based on a simple example.

Though the example has toy character, it should suffice to demonstrate the wide range of manipulation possibilities given by an internal integrated graph representation for the enhancement of urban object models. Since TGraph technology is easily applicable to graphs containing millions of elements, the approach scales to a wide range of applications.

The integrated model was developed in the context of a project for object-recognition (Falkowski et al., 2009). It forms the basis for the application of efficient graph-matching algorithms in this context.

The integrated model schema is still under construction. But it is easily modifiable and each of the three parts can be replaced by different variants. Further goals are the enhancement of the schema for the full *CityGML base profile* (CityGML[full]) and the support for other urban object description languages, like *KML/COLLADA* (section 1.1). Here the tasks are the change and enlargement of the integrated model schema and the adaption of all existing processing components. Some of the described activities could be splitted to more processing steps. A lot of them can be composed to interesting combined processing activities. And there could even be interactive processing components.

Further research topics could be the supplement of more complex model parts to an existing integrated model or the integration of two different integrated models. Another interesting field is the inference of semantics from geometric, topological and/or appearance information.

ACKNOWLEDGEMENTS

This work has been carried out in close cooperation with Peter Decker, Dietrich Paulus and Stefan Wirtz from the Work Group Active Vision as well as Lutz Priese and Frank Schmitt and from the Laboratory Image Recognition, both at the University of Koblenz-Landau.

REFERENCES

Cox, S., Daisey, P., Lake, R., Portele, C. and Whiteside, A., 2001. OpenGIS Geography Markup Language (GML) Implementation Specification. Technical Report 3.1.1, Open Geospatial Consortium, Inc.

Czarnecki, K. and Eisenecker, U. W., 2000. Generative Programming: Methods, Tools and Applications. ACM Press/Addison-Wesley Publishing Co., New York, NY, USA.

Ebert, J., Riediger, V. and Winter, A., 2008. Graph Technology in Reverse Engineering, The TGraph Approach. In: R. Gimnich, U. Kaiser, J. Quante and A. Winter (eds), 10th Workshop Software Reengineering (WSR 2008), GI Lecture Notes in Informatics, Vol. 126, GI, Bonn, pp. 67–81.

Falkowski, K., Ebert, J., Decker, P., Wirtz, S. and Paulus, D., 2009. Semiautomatic generation of full CityGML models from images. In: Geoinformatik 2009, ifgiPrints, Vol. 35, Institut für Geoinformatik Westfälische Wilhelms-Universität Münster, Osnabrück, Germany, pp. 101–110.

Foley, J. D., van Dam, A., Feiner, S. K. and Hughes, J. F., 1990. Computer Graphics. Principles and Practice. Addison Wesley.

Gamma, E., Helm, R., Johnson, R. and Vlissides, J., 1995. Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA.

Groeger, G., Kolbe, T. H., Czerwinski, A. and Nagel, C., 2008. OpenGIS City Geography Markup Language (CityGML) Encoding Standard. Technical Report 1.0.0, Open Geospatial Consortium Inc.

Herring, J., 2001. Topic 1: Feature Geometry (ISO 19107 Spatial Schema). Technical Report 5.0, Open Geospatial Consortium Inc.

Muller, D. E. and Preparata, F. P., 1978. Finding the Intersection of two Convex Polyhedra. Theor. Comput. Sci. 7, pp. 217–236.

Pohl, K., Böckle, G. and van der Linden, F., 2005. Software Product Line Engineering. Springer, Berlin.

A PROOF OF CONCEPT OF ITERATIVE DSM IMPROVEMENT THROUGH SAR SCENE SIMULATION

D. Derauw

Signal and Image Centre – Royal Military Academy, Renaissance Av., 1000 Brussels, Belgium - dderauw@elec.rma.ac.be
Centre Spatial de Liège – Université de Liège, Avenue du Pré Aily, 4031 Angleur, Belgium - dderauw@ulg.ac.be

KEY WORDS: SAR, Scene, Simulation, DEM, reconstruction

ABSTRACT:

In Very High Resolution (VHR) Synthesis Aperture Radar (SAR) context, very fine and accurate georeferencing and geoprojection processes are required. Both operations are only applicable if accurate local heights are known. 3D information may be derived from SAR interferometry (InSAR), But in VHR context, InSAR reveals to be inaccurate mostly due to phase unwrapping problems and to phase/height noise. Generated InSAR Digital Surface Models (DSM) can only be considered as a first good approximation of the observed surface. Therefore, we proposed to start from the InSAR DSM, to project it on ground range on a given datum, to model the observed scene using this projected DSM, then to simulate in slant range the intensity image issued from this structure model. Comparison between simulated and observed intensity image can then be used as a criterion to modify and improve the considered underlying DSM.
In this paper, we present the different steps of the proposed approach and results obtained so far, showing that the proposed process can be run iteratively to modify the DSM and reach a stable solution.

1. INTRODUCTION

A cooperation programme named ORFEO (Optic and Radar Federated Earth Observation) was set up between France and Italy to develop an Earth observation dual system, optic and radar, with metric resolution. Italy is in charge of the radar component (COSMO-Skymed), and France of the optic component (PLEIADES).

Beside ORFEO, an accompanying programme was set-up to prepare the use and joint exploitation of images that will be provided from this satellites constellation. In the frame of this accompanying programme, the Belgian Science Policy (BelSPo) is financing the EMSOR project aiming at performing man-made object detection for urban map updating using VHR SAR and optical data.

While such objective is well addressed in the optical imagery, this topic stays highly challenging in SAR imagery due to inherent peculiarities of SAR acquisition and imaging mode. Main obstacles are geometrical on one side and linked to SAR signal content on the other side. Geometrical deformation specific to SAR systems, i.e. layover, foreshortening, shadowing, make man-made structures appearing very differently in shape with respect to their appearance in optical imagery (Balz T. 2003).

Specificities of SAR signal, mainly speckle, radar cross section dependence with incidence angle and multiple reflection processes make identical objects appear sufficiently differently to compromise, or make inoperative, classical detection techniques applicable in optical imagery. Man-made structures detection in SAR images based on speckle filtering followed by image segmentation is not applicable as such. Classification is often considered as a first processing step that, combined with other information layers, is used in higher level processing for fine Digital Surface Model (DSM) extraction and man-made structure detection (Tison et al. 2007, Thiele et al. 2007). SAR scene simulation was also proposed to help in fine georeferencing process (Blaz T. 2006) or to iteratively steer building structures detection and identification (Soerger et al., 2003).

Similarly, in this paper, we propose an iterative way to improve a seed DSM that is obtained through classical Interferometric processing of single pass VHR SAR data. We developed a basic SAR intensity image simulator adapted to very high resolution. This one is then used to improve our seed DSM, comparing the simulated image in intensity with the detected one and using this comparison to perform blind DSM corrections without any a priori knowledge of the underlying urban structure.

The proposed approach is justified by the fact that classical interferometric SAR (InSAR) is showing its limits in the VHR context. Therefore, on-ground projected InSAR DSM can be considered as a first approximation of the 3D observed surface and be used as a seed DSM to be improved.

The main aim being man-made structure detection, improvement means here reaching a DSM representation allowing better detection and localisation of searched structures.

This paper describes first results obtained and choices that have been made up to now to assess the validity of the proposed iterative process. Our first aim was to perform a proof of concept of the proposed approach, i.e. DSM improvement based on iterative comparison between a simulated and detected SAR intensity image.

2. TEST SITE AND SEED DSM

2.1 Data set description

To generate our seed DSM, we are using a VHR InSAR pair acquired in February 2006 above Toulouse (France) by the RAMSES X-band sensor (Dupuis et al. 2000). Resolution cell dimensions are 0.55m in azimuth by 0.35m in slant range. We

limited the data to a sub set of approximately 2000x2000 pixels at full slant range-azimuth resolution. This subset contains both man-made structures and open vegetated areas.

2.2 Seed DSM generation

Seed DSM is first generated in Slant range projection using interferometric processing. Working at Very High Resolution may induce some local problems mainly in the phase unwrapping process.

Man-made structures and more generally all features observed at VHR induce rapid height variation with respect to the resolution cell dimension. Since working at full resolution, these rapid height variations combined with phase noise induce in turn high spatial frequencies in the interferometric phase, making the phase unwrapping process potentially difficult even if the ambiguity of altitude is high compared to buildings heights. The generated InSAR DSM contains some small holes made of local DSM areas unwrapped independently. Figure 1 shows the amplitude image of the sample data set in slant range with the derived DSM.

Figure 1: Data set and corresponding seed DSM in slant range

After the phase unwrapping process, the seed DSM is still in slant range azimuth geometry. Before being considered as the seed DSM to be iteratively improved, it must be geo-referenced and projected in a convenient geometry.

A convenient geometry is a projection within which further processing for man-made structure detection, localisation and identification will be feasible but also a projection geometry within which SAR scene simulation will stay easy to model.

Considering first that man-made structures have no preferential orientation within an observed scene, there is no peculiar advantage of using a specific geographic or cartographic projection rather than another. Therefore, with respect to man-made structure detection, the important point is to work on geo-projected data to get rid of geometrical aspects linked to the slant range geometry. Consequently, working within a given geographic or cartographic projection is of no peculiar importance.

Considering SAR scene simulation, we need a projection geometry allowing to easily model radar wave interaction with the observed scene. Interactions taken into account here are purely geometrical (ray tracing). At the present time, we do not intent to take a local backscattering coefficient into account, even if possibilities to integrate it in the model will be envisioned at each implementation steps.

Based on these considerations, the ground range projection was chosen. This geometry is certainly the simplest to be considered for SAR scene simulation, while, with respect to man-made structure localization, it is not necessarily the most convenient. Therefore, when performing ground range projection, geo-referencing of each point in terms of longitude and latitude will be saved to allow further projection in any geographic or cartographic reference system.

2.3 Structure definition

Once geo-projected, the seed DSM must be used to define a structure that in turn will be used to model the backscattered SAR signal and simulate the detected SAR scene. Therefore, structure definition depends mainly on the way the simulation process is envisioned. The basic idea is to associate to each point of the DSM, a value that is proportional to the backscattered energy, giving then a peculiar weight to each point. Next, this map of backscattered energy will simply be back-projected in slant range to generate a simulated image.

In a first approach, we simply aimed at considering non-coherent dihedral reflection as the main backscattering process to be taken into account.

2.3.1 Dihedral structures: Once more, for the sake of simplicity and in order to allow us to first perform a proof of concept, we choose to use directly the DSM as the structure itself. Simply, two consecutive heights are used to define a dihedral. The DSM is considered sequentially, azimuth lines by azimuth lines, and within a line, heights are considered sequentially with increasing ground range. If a given height is greater than the preceding one, a dihedral structure can be defined (fig 2).

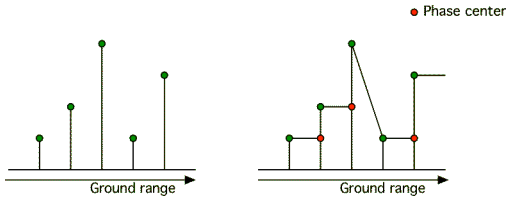


Figure 2: DSM height interpretation

The part of the incident beam intercepted by a dihedral structure will be fully backscattered toward the beam source. Therefore, the backscattered energy will be proportional to the square of the aperture of the considered dihedral structure; the aperture being the hypotenuse of the illuminated part of the dihedral.

Any entering beam in the dihedral follows an optical path of the same length. Therefore, all entering beams will be imaged as localized at the phase centre of the dihedral. Since we are working azimuth lines by azimuth lines, our basis structure is defined in 2D and the phase centre is localized at the intersection of the local horizontal and the local vertical of the considered point.

If we consider two consecutive points of our DSM along a ground range line having respectively heights h_{i-1} and h_i, a dihedral structure will basically be defined if $h_i > h_{i-1}$; its phase centre will be localized at ground range coordinate of h_i with local height h_{i-1} and have a weight proportional to its aperture.

2.3.2 Overestimation: Normally, the aperture of a dihedral should be computed taking into account shadowing of preceding dihedrals, if any, and be computed with respect to the height difference or with respect to the base, whatever the one is limiting the aperture the first.

At the present time, we decided to compute the aperture in the simplest way possible to rapidly have a functioning iterative process. Improvements of the structure model will be considered at a later stage. Therefore, apertures are computed directly from the local height difference and from the local incidence angle, not taking into account the base of the dihedral (fig. 3). This can lead to an overestimation of the dihedral aperture.

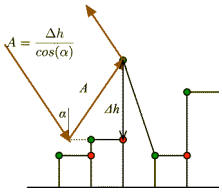


Figure 3: Basic model of dihedral back-scattering

2.3.3 Dihedral aperture / surface scattering limit: If dihedral backscattering process may be considered as predominant in the presence of man-made structures in terms of backscattered energy, surface scattering must also be taken into account for open areas that are also well present at VHR.

Considering only dihedral backscattering process tends to segment the structure; each time a local height is lower than the preceding one, the aperture, and so the backscattered energy, will be considered as null.

Therefore, we determined a simple height variation limit above which, we consider that dihedral backscattering process occurs and below which, surface backscattering is taking place. The chosen limit is simply the one inducing layover. If the local height difference induces layover, we consider that we have to deal with a dihedral structure, if not, we consider we have to deal with an elementary surface (fig. 4 & 5).

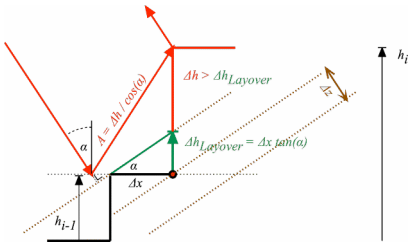


Figure 4: Dihedral structure – surface scattering limit

Above the layover limit, the weight of a point will be calculated as its dihedral aperture. Below this limit, surface scattering will be considered.

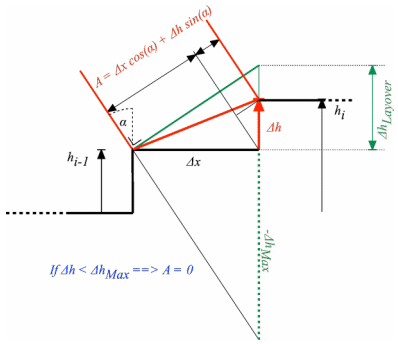


Figure 5: Surface scattering component

In case of surface scattering, not taking into account a specific local backscattering coefficient, the backscattered energy is taken as proportional to the beam section intercepted by the considered pixel. In place of dihedral aperture, we can thus speak in terms of pixel aperture (fig. 5).

As depicted in figure 5, the intercepted beam section will decrease with the height variation between two pixels up to zero when the shadowing limit is reached.

In terms of backscattered energy, surface backscattering process has a much lower weight than dihedral reflection. Therefore, in practice, a fix coefficient will be applied between both aperture types. At this level, a local backscattering coefficient and/or an emission diagram at pixel level depending on the local slope and on the local incidence should be considered as supplementary weighting factors.

It follows that for a given DSM we define a structure that allows taking into account two backscattering process: dihedral and surface, each with a different weight. Once again, for the sake of simplicity, the current model attributes the computed pixel aperture to the point located at the current position i with height h_i as if the point was a phase centre, even if considering surface scattering.

Consequently, our model defines only point scatterers located on a ground range – azimuth mesh for which height are issued from the projected DSM that must be updated and improved iteratively. At each of these point scatterer position, we will consider we have a point scatterer response whose relative intensity will be determined by the computed aperture.

3. BACK AND FORTH REFERENCING PROCESS

The back and forth referencing and projection processes we have implemented were specifically developed for space-borne sensors. Therefore, no flight motion compensation is considered here. Referencing is thus deduced considering an analytical trajectory of the sensor on its orbit, a fix Doppler cone for the whole scene and a reference geoid (WGS84).

3.1 Ground range referencing

Existing geo-referencing processes allows finding geocentric Cartesian coordinates of a given point in slant range coordinate of know height above the geoid. This geocentric coordinate can then be translated in geodetic coordinate and converted in longitude latitude on the considered datum. Therefore, there is an analytical link between the slant range coordinates of a point of known altitude and its coordinate in a geocentric Cartesian system or in a given cartographic system.

The ground range coordinate of a point given in slant range is defined as the length of a curve segment, which is the intersection between the chosen geoid and the Doppler cone, the length being calculated through integration from the minimum slant range point to the considered point. This integration makes the reverse calculation complicate. Therefore, in the process of calculating the ground range coordinate of a point, this latter one is first geo-referenced on the considered geoid, in longitude - latitude coordinate. This allows building a map linking ground range coordinates with geographical coordinates. This map is then fitted by a second order polynomial for both the longitude and the latitude.

3.2 Slant range referencing

To complete the back and forth projection process, we also need a computational way to reference a point, given in ground range coordinate, back to slant range coordinate. We simply use the second order polynomial linking a ground range position to its longitude – latitude coordinates to find back its geographical position. These geographical coordinates are then converted in Cartesian coordinates in the Earth center coordinate system and the range is derived computing the distance between the position of the sensor on its orbit and the Cartesian coordinate of the considered point.

Special attention was drawn to this back and forth referencing process to ensure reliability and accuracy in accordance with VHR context. In practice, mathematically speaking, the referencing process can easily reach centimeter precision.

4. SAR SCENE SIMULATION

4.1 From aperture to simulated intensity

As explained previously, from a DSM projected in ground range, we build a structure allowing to define either dihedral or surface backscattering. To each point on the ground range sampling grid, we associate what we have called an aperture, which is an evaluation of the incident energy intercepted by the dihedral or the considered surface element. Therefore, from a DSM, we build what might be called an aperture map.

Each point of the ground range mesh is thus considered as a point scatterer to which is associated a point scatterer response backscattering an energy proportional to the incident one.

The ground range mesh is then referenced back in slant range and the corresponding projection map is built. For each destination point in slant range, the projection map contains the coordinates of all intervening points in ground range coordinate. Intervening points are those that have to be taken into account to extrapolate the projected value at the considered slant range – azimuth position. After this step, we know the location of the centre of each intervening point scatterer response with respect to a given slant range –azimuth position.

The pixel at that position receives from a given point scatterer response, an energy that is the integral of the impulse response, limited to the slant range pixel area. This integral will be the weight attributed to the contribution of the considered point scatterer response. The simulated energy is obtained summing all contributions of all intervening point scatterers responses for a given slant range pixel.

In SAR, the impulse response or point scatterer response in slant range – azimuth is a sinc-like function generally approximated by a sinc function (Bamler 1993). In terms of energy, we thus deal with a square sinc function and our apertures map in slant range must then be considered as a mesh of square sinc functions of different heights.

In practice, computing the integral of bi-dimensional squared sinc function on a given interval is highly complex. Therefore, we approximate our point scatterer response by a Gaussian having the same width at half maximum. The advantage is that calculating the integral of a bi-dimensional Gaussian on a given interval is straightforward (Fig. 6). One drawback is that strong side lobes issued from dihedral backscattering process are not modelled.

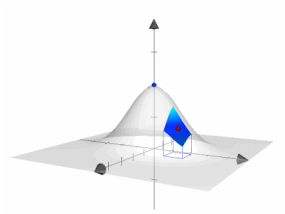


Figure 6: Integration of an approximated point scatterer response limited to a target slant range pixel

Figure 7 shows the square root of the simulated image obtained in slant range starting from our seed DSM given in ground range and following the whole procedure described here-above. The real detected SAR image is shown on the right of the figure for qualitative comparison. For the sake of clarity and to improve contrast, the square roots of the simulated intensities are represented.

If, from a macroscopic point of view, similar structures are roughly observable, the simulated image does show a level of details very far from the one of the detected SAR image. Reasons of having apparently so poor results may have three distinct origins: the seed DSM quality, the structure model used for estimating the local backscattered energy and the used parameters.

Figure 7: Simulated SAR scene based on seed DSM structure

When projecting the InSAR DSM onto ground range to build the seed DSM, available parameters are on-ground resolution cell dimension, semi-major and semi-minor axis of the ellipse used to find intervening points, weighting method and interpolation method. These parameters have great influence on the smoothing and the quality of the seed DSM.

In the reverse process, when referencing the backscattering structure toward slant range, parameters are the azimuth and slant-range resolution to determine the point scatterer response width, semi-major and semi-minor axis of the ellipse used to find intervening points and the resolution cell dimension of the targeted simulated image.

5. DSM ITERATIVE MODIFICATIONS

At this stage, we have the tools required to link slant range and ground range geometries allowing a back and forth process. The DSM in it self is now in ground range geometry and allows generating a simulated SAR intensity image in slant range geometry to be compared to the really detected one.

For a given point in the simulated image, we have the mapping that lists the points of the seed DSM with their respective weights intervening in the simulation. Conversely, we also have the reverse mapping that, for a given point of the seed DSM, lists points in the simulated image into which the considered DSM point intervene with respective weights. It is this reverse mapping that is used in the DSM modification process.

5.1 Normalisation

To be correct, the simulated image is considered as being an intensity image within an unknown proportionality factor. Before being usable as a valid scene for comparison with the really detected image, the simulated one must be normalized. The normalisation factor is simply the ratio of the integral of the backscattered energy measured in Digital Numbers (DN) in the detected image to the integral of simulated energy.

After normalization, both images represent the same energy globally backscattered by the whole scene, which allows a comparison on a point-by-point basis.

5.2 Improvement criterion

The chosen comparison criterion is simply the local energy ratio. In other words, if the detected energy is higher than the simulated one, the underlying aperture used for the simulation must be increased proportionally.

In the facts, several apertures intervene with different weights in the simulation of a point. Therefore, we work in the reverse way, using the reverse mapping. For a given point of the DSM, the reverse mapping gives us the list of all simulated point into which the considered DSM point intervene with corresponding weights. Consequently, we perform a weighted average of the energy ratios on these simulated and detected points. This weighted average gives us the proportionality factor that should be applied to the underlying aperture.

Whatever the considered backscattering process, apertures are proportional to the local height difference between consecutive points in ground range. Therefore, the proportionality factor can directly be applied to the local height of the DSM under concern.

To summarize, DSM points are corrected sequentially in ground range using a weighted average of intensity ratio calculated on several points in slant range – azimuth. These slant range points are those for which the DSM point under concerns plays a role through the aperture it generates.

5.3 Iterative process

When the corrected DSM is issued, the whole process can be reiterated, starting anew from this new DSM. This latter one will thus be used to compute a new aperture structure and to compute the ground to slant range projection mapping.

The mapping will be used in an additive way to generate a simulated SAR intensity image, which, after normalization with respect to the detected one, will be used for DSM improvement. The simulated scene shown on figure 7 can thus be considered as the first iteration of the iterative process described here above.

Figure 8 shows the second iteration of the simulated scene so obtained. The simulated scene appears still of poor quality, but some structures appears more clearly. Corrections with respect to the first iteration are quite important, and mainly a first segmentation between highly urbanized areas and open areas has roughly been made.

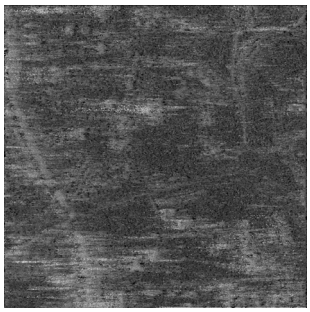


Figure 8: Simulated SAR scene after 2 iterations

From a computational point of view, in debug mode, one iteration takes about 4 minute a run for a seed DSM of about 2000x2000 points. This computation time being reasonable, up to 25 iterations have been performed. Figure 9 shows results obtained after 4 and 12 iterations. Figure 10 shows the last iteration along with the really detected scene.

Figure 9: Simulated SAR scene obtained after 4 (left) and 12 (right) iterations

Figure 10: Simulated SAR scene obtained after 25 iterations (left) and really detected one (right)

Clearly, the iterative process converges toward a stable simulation. Qualitatively, convergence appears to be more rapid between the few firsts iterations, while improvement between iteration 12 and 25 becomes less evident. Therefore, the proposed process seams to converge monotonically toward a solution.

It must be noted that the iterative process converges toward a solution that is linked to the underlying aperture model, which in turn, is linked to an improved DSM. Our “improved” DSM is thus “**one possible representation of the observed surface**”. This possible representation of the observed surface is the one that can be obtained with the developed structure model and using a peculiar set of parameters.

Figure 11 shows in parallel, the seed DSM computed in ground range along with the improved one obtained after 25 iterations.

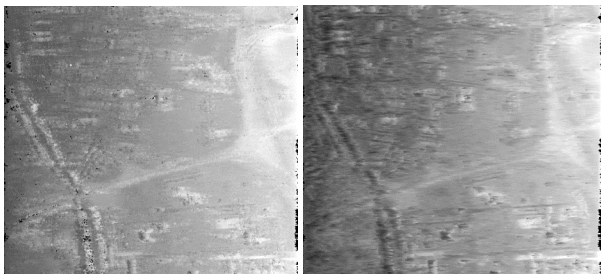


Figure 11: Seed and improved DSM

While the simulated SAR scene is clearly improved after 25 iterations, improvement is less evident observing the obtained DSM.

Figure 12 represents a DSM sample line, in ground range, before and after improvement. Globally, we observe that the modified DSM appears less noisy and more structured. At this stage, it is difficult to assert if the reached structure is a correct representation of the observed scene and if it can be used in man-made structure detection or identification. But, we can conclude that the achieved structure, together with the proposed model and the used parameter set, allows simulating a SAR intensity image close to the really detected one.

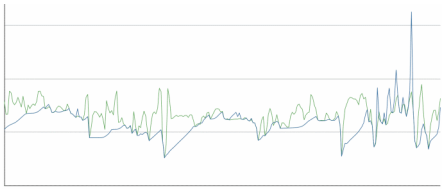


Figure 11: DSM sample line before (green) and after (blue) improvement

Obtaining a DSM representation closer to the observed one will require testing the influence of all parameters as also improving our simplistic model. But, the main point is that we performed a proof of concept of the proposed principle: “Iterative DSM improvement through SAR scene simulation and comparison with observed one”.

Since the proposed method is global and does not require any a priori knowledge on buildings shapes and orientation, it can be envisioned as a first improvement of the DSM to be used in more sophisticated and context-based man-made structure detection techniques.

Nevertheless, if stable, the reached simulated SAR intensity image stays, for the moment, still far from the really detected SAR intensity image. We have well concentrated the energy where it should, but still not with the degree of details offered by the real data. One must thus keep in mind that the obtained improved DSM is just one possible representation of the observed scene. Other representations are possible provided simulation model and set of parameters that are used are optimized

6. CONCLUSIONS

We developed the tools required for simulating a SAR intensity image in slant range geometry starting from a seed DSM given in ground range and issued from InSAR processing.

Our objective was first to perform a proof of concept, showing that in its principle, it is possible to perform an iterative improvement of a seed DSM by simulation of SAR intensity image in slant range – azimuth projection and comparison with the corresponding detected one. Therefore, we developed a simplistic model allowing to associate a backscattered energy to ground range – azimuth resolution cells with respect to local heights.

Effort was principally put on the reliability and accuracy of back and forth referencing and projection processes.

Clearly, the proof of concept is performed: comparing simulated and detected backscattered energy in slant range allows correcting iteratively the underlying DSM.

The process converges monotonically toward a DSM structure that is thus one possible representation of the observed scene. Monotonic convergence shows that the obtained solution is stable and is, in itself, the result that had to be obtained to validate the proposed iterative process.

Complementary analysis must be performed to assess if the derived DSM can efficiently be used for man-made structures detection.

7. REFERENCES

Balz T, Haala N. (2003), *SAR-based 3D reconstruction of complex urban environments*, The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Vol. 34, Part 3/W13, pp.181-185

Balz, T (2006), *Automated CAD model-based geo-referencing for high-resolution SAR data in urban environments.* Radar, Sonar and Navigation, IEE Proceedings - Vol. 153(3), June 2006, pp. 289 - 293

Bamler. R and Schattler B. (1993), *SAR data acquisition and image formation*. In: Schreier G. (ed.) *SAR geocoding: data and systems*.. Wichmann, Karlsruhe, pp. 53-101.

Dupuis X., Dupas J., Oriot H. (2000) 3D extraction from interferometric high resolution SAR images using the RAMSES sensor, PROC. 3rd European Symposium on Synthetic Aperture Radar, EUSAR'2000, München (Germany), VDE, pp. 505-507

Soerger U., Thoennessen U., Stilla U. (2003), *Reconstruction of buildings from interferometric SAR data of built-up areas*. The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Vol. 34, Part 3/W13, pp. 59-64

Thiele, A.; Cadario, E.; Schulz, K.; Thonnessen, U.; Soergel, U (2007), *Building Recognition From Multi-Aspect High-Resolution InSAR Data in Urban Areas*, IEEE Transactions on Geoscience and Remote Sensing, 45(11), pp. 3583 - 3593

Tison C., Tupin F., Maitre H. (2007), *A fusion scheme for joint retrieval of urban height map and classification from high-resolution interferometric SAR images*, IEEE Tranactions on Geosciences and Remote Sensing,, vol. 45(2), pp. 496-50

COMPETING 3D PRIORS FOR OBJECT EXTRACTION IN REMOTE SENSING DATA

Konstantinos Karantzalos and Nikos Paragios

Ecole Centrale de Paris
Grande Voie des Vignes, 92295
Chatenay-Malabry, France
{konstantinos.karantzalos, nikos.paragios}@ecp.fr
http://users.ntua.gr/karank/Demos.html

Commission III

KEY WORDS: Computer Vision, Pattern Recognition, Variational Methods, Model-Based, Evaluation, Voxel-Based

ABSTRACT:

A recognition-driven variational framework was developed for automatic three dimensional object extraction from remote sensing data. The essence of the approach is to allow multiple 3D priors to compete towards recovering terrain objects' position and 3D geometry. We are not relying, only, on the results of an unconstrained evolving surface but we are forcing our output segments to inherit their 3D shape from our prior models. Thus, instead of evolving an arbitrary surface we evolve the selected geometric shapes. The developed algorithm was tested for the task of 3D building extraction and the performed pixel- and voxel-based quantitative evaluation demonstrate the potentials of the proposed approach.

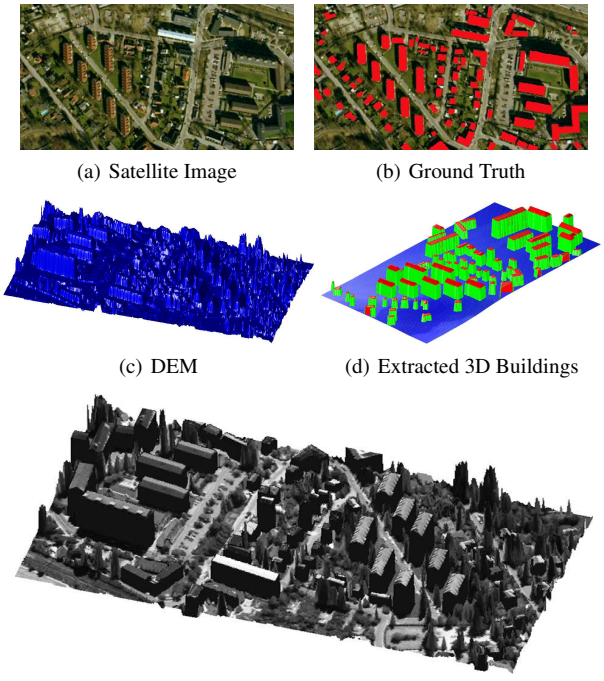
1 INTRODUCTION

Although, current remote sensing sensors can provide an updated and detailed source of information related to terrain analysis, the lack of automated operational procedures regarding their processing impedes their full exploitation. By using standard techniques based, mainly, on spectral properties, only the lower resolution earth observation data can be effectively classified. Recent automated approaches are not, yet, functional and mature enough for supporting massive processing on multiple scenes of high- and very high resolution data.

On the other hand, modeling urban and peri-urban environments with engineering precision, enables people and organizations involved in the planning, design, construction and operations lifecycle, in making collective decisions in the areas of urban planning, economic development, emergency planning, and security. In particular, the emergence of applications like games, navigation, e-commerce, spatial planning and monitoring of urban development has made the creation and manipulation of 3D city models quite valuable, especially at large scale.

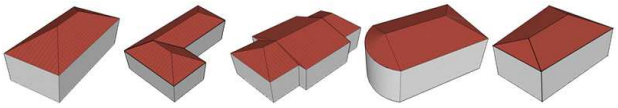
In this perspective, optimizing the automatic information extraction of terrain features/objects from new generation satellite data is of major importance. For more than a decade now, research efforts are based on the use of a single image, stereopairs, multiple images, digital elevation models (DEMs) or a combination of them. One can find in the literature several model-free or model-based algorithms towards 2D and 3D object extraction and reconstruction [(Hu et al., 2003),(Baltsavias, 2004),(Suveg and Vosselman, 2004),(Paparoditis et al., 2006),(Drauschke et al., 2006),(Rottensteiner et al., 2007),(Sohn and Dowman, 2007),(Verma et al., 2006),(Lafarge et al., 2007),(Karantzalos and Paragios, 2009) and the references therein]. Despite this intensive research, we are, still, far from the goal of the initially envisioned fully automatic and accurate reconstruction systems (Brenner, 2005),(Zhu and Kanade (Eds.), July, 2008),(Mayer, 2008). Processing remote sensing data, still, poses several challenges.

In this paper, we extend our recent 2D prior-based formulations (Karantzalos and Paragios, 2009) aiming at tackling the problem of automatically and accurately extracting 3D terrain objects from optical and height data. Multiple 3D competing priors are considered transforming reconstruction to a labeling and an estimation problem. In such a context, we fuse images and DEMs towards recovering a 3D prior model. We are experimenting with buildings but, similarly, any other terrain object can be modeled. Our formulation allows data with the higher spatial resolution to constrain properly the footprint detection in order to achieve the optimal spatial accuracy (Figure 1). Therefore, we are proposing a variational functional that encodes a fruitful synergy between observations and multiple 3D grammar-based models. Our models refer to a grammar, which consists of typologies of 3D shape priors (Figure 2). In such a context, firstly one has to select the

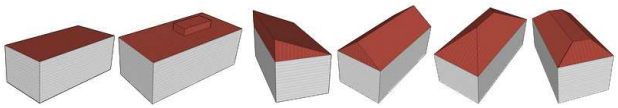


(a) Satellite Image (b) Ground Truth (c) DEM (d) Extracted 3D Buildings (e) Reconstructed Scene

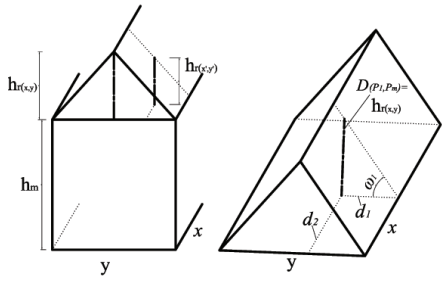
Figure 1: 3D Building Extraction through Competing 3D Priors



(a) Prior Building Models ($\tilde{\Phi}_{\mathbf{i},\mathbf{j}}$): $\mathbf{i}$ determines the shape of the footprint and $\mathbf{j}$ the roof type



(b) The family $\tilde{\Phi}_{1,\mathbf{j}}$ which has a rectangular footprint ($\mathbf{i} = 1$).



(c) Building's main height h_m and roofs height $h_r(x, y)$

Figure 2: Hierarchical Grammar-Based 3D Prior Models. The case of Building Modeling: Building's footprint is determined implicitly from the E_{2D}. h_m and $h_r(x, y)$ are recovered for every point (E_{3D}) and thus all the different type of roofs $\mathbf{j}$ are modeled.

most appropriate model and then determine the optimal set of parameters aiming to recover scene's geometry (Figure 1). The proposed objective function consists of two segmentation terms that guide the selection of the most appropriate typology and a third DEM-driven term which is being conditioned on the typology. Such a prior-based recognition process can segment both rural and urban regions (similarly to (Matei et al., 2008)) but is able, as well, to overcome detection errors caused by the misleading low-level information (like shadows or occlusions), which is a common scenario in remote sensing data.

Our goal was to develop a single generic framework (with no step-by-step procedures) that is able to efficiently account for multiple 3D building extraction, no matter if their number or shape is *a priori* familiar or not. In addition, since usually for most sites multiple aerial images are missing, our goal was to provide a solution even with the minimum available data, like a single panchromatic image and an elevation map (produced either with classical photogrammetric multi-view stereo techniques either from LIDAR or INSAR sensors), contrary to approaches that were designed to process multiple aerial images or multispectral information and cadastral maps (like in (Suveg and Vosselman, 2004),(Rottensteiner et al., 2007),(Sohn and Dowman, 2007)), data which much ease scene's classification. Doing multiview stereo, using simple geometric representations like 3D lines and planes or merging data from ground sensors was not our interest here. Moreover, contrary to (Zebedin et al., 2008), the proposed, here, variational framework does not require as an input dense height data, dense image matching processes and *a priori* given 3D line segments or a rough segmentation.

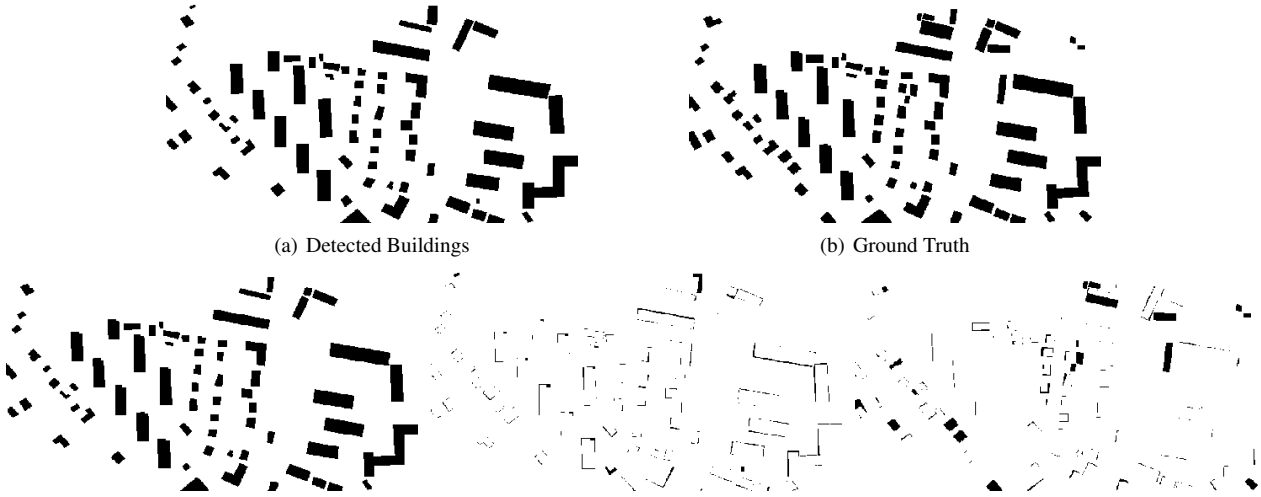
2 MODELING TERRAIN OBJECTS WITH 3D PRIORS

Numerous 3D model-based approaches have been proposed in literature. Statistical approaches (Paragios et al., 2005), aim to describe variations between the different prior models by measuring the distribution of the parameter space. These models are capable to model building with rather repeating structure and of limited complexity. In order to overcome this limitation, methods using generic, parametric, polyhedral and structural models have been considered (Jaynes et al., 2003),(Kim and Nevatia, 2004),(Suveg and Vosselman, 2004),(Dick et al., 2004),(Wilczkowiak et al., 2005),(Forlani et al., 2006),(Lafarge et al., 2007). The main strength of these models is their expressional power in terms of complex architectures. On the other hand, inference between the models and observations is rather challenging due to the important dimension of the search space. Consequently, these models can only be considered in a small number. More recently, procedural modeling of architectures was introduced and vision-based reconstruction in (Muller et al., 2007) using mostly facade views. Such a method recovers 3D using an L-system grammar (Muller et al., 2006) that is a powerful and elegant tool for content creation. Despite the promising potentials of such an approach, one can claim that the inferential step that involves the derivation of models parameters is still a challenging problem, especially when the grammar is related with the building detection procedure.

Hierarchical representations are a natural selection to address complexity while at the same time recover representations of acceptable resolution. Focusing on buildings, our models involve two components, the type of footprint and the type of roof (Figure 2). Firstly, we structure our prior models space $\tilde{\Phi}$ by ascribing the same pointer $\mathbf{i}$ to all models that belong to the family with the same footprint. Thus, all buildings that can be modeled with a rectangular footprint are having the same index value $\mathbf{i}$. Then, for every family (i.e. every $\mathbf{i}$) the different types of building tops (roofs) are modeled by the pointer $\mathbf{j}$ (Figure 2b) Under this hierarchy $\tilde{\Phi}_{\mathbf{i},\mathbf{j}}$, the priors database can model from simple to very complex building types and can be easily enriched with more complex structures. Such a formulation is desirously generic but forms a huge search space. Therefore, appropriate attention is to be paid when structuring the search step.

Given the set of footprint priors, we assume that the observed building is a homographic transformation of the footprint. Given, the variation of the expressiveness of the grammar, and the degrees of freedom of the transformation, we can now focus on the 3D aspect of the model. In such a context, only building's main height h_m and building's roof height $h_r(x, y)$ at every point need to be recovered. The proposed typology for such a task is shown in Figure 2. It refers to the rectangular case but all the other families can respectively be defined. More complex footprints, with usually more than one roof types, are decomposed to simpler parts which can, therefore, similarly recovered. Given an image $\mathcal{I}(x, y)$ at domain (bounded) $\Omega \in R^2$ and an elevation map $\mathcal{H}(x, y)$ -which can be seen both as an image or as a triangulated point cloud- let us denote by h_m the main building's height and by P_m the horizontal building's plane at that height. We proceed by modeling all building roofs (flat, shed, gable, etc.) as a combination of four inclined planes. We denote by P_1, P_2, P_3 and P_4 these four roof planes and by $\omega_1, \omega_2, \omega_3$ and ω_4, respectively, the four angles between the horizontal plane h_m and each inclined plane (Figure 2). Every point in the roof rests strictly on one of these inclined planes and its distance with the horizontal plane is the minimum compared with the ones formed by the other three planes.

With such a grammar-based description the five unknown parameters to be recovered are: the main height h_m (which has a constant value for every building) and the four angles ω. In this way all -but two- types of buildings tops/roofs can be modeled. For example, if all angles are different we have a totally dissymmetric roof (Figure 2b - $\tilde{\Phi}_{\mathbf{1},\mathbf{5}}$), if two opposite angle are zero we have a



(a) Detected Buildings (b) Ground Truth

(c) Horizontal True Positives (HTP) (d) Horizontal False Positives (HFP) (e) Horizontal False Negatives (HFN)

Figure 3: Horizontal Qualitative Evaluation: The recognition-driven process efficiently detects, in an unsupervised manner, scene buildings and recovers their 3D geometry.

gable-type one ($\tilde{\Phi}_{\mathbf{1},\mathbf{4}}$) and if all are zero we have a flat one ($\tilde{\Phi}_{\mathbf{1},\mathbf{1}}$). The platform and the gambrel roof types can not be modeled but can be easily derived in cases where the fit energy metric is assumed on local minima. The platform one ($\tilde{\Phi}_{\mathbf{1},\mathbf{2}}$), for instance, is the case where all angles have been recovered with small values and a search around their intersection point will estimate the dimensions of the rectangular-shape box above main roof plane P_m. With the aforementioned formulations, instead of searching for the best among $\mathbf{i}\mathbf{x}\mathbf{j}$ (e.g. $5\text{x}6 = 30$) models, their hierarchical grammar and the appropriate defined energy terms (detailed in the following section) are able to cut down effectively the solutions space.

3 MULTIPLE 3D PRIORS IN COMPETITION EXTRACTING MULTIPLE OBJECTS

Let us consider an image ($\mathcal{I}$) and the corresponding digital elevation map ($\mathcal{H}$). In such a context, one has to separate the desired for extraction objects from the background (natural scene) and, then, determine their geometry. The first segmentation task is addressed through the deformation of a initial surface $\phi : \Omega \rightarrow \mathcal{R}^+$ that aims at separating the natural components of the scene from the man-made parts. Assuming that one can establish correspondences between the pixels of the image and the ones of the DEM, the segmentation can be solved in both spaces through the use of regional statistics. In the visible image we would expect that buildings are different from the natural components of the scene. In the DEM, one would expect that man-made structures will exhibit elevation differences from their surroundings. Following the formulations of (Karantzalos and Paragios, 2009), these two assumptions can be used to define the following segmentation function

$$E_{seg}(\phi) = \int |\nabla\phi(\mathbf{x})|\, d\mathbf{x} + \int_{\Omega} H_\epsilon(\phi)\, r_{obj}\left(\mathcal{I}(\mathbf{x})\right) + [1 - H_\epsilon(\phi)]\, r_{bg}\left(\mathcal{I}(\mathbf{x})\right)\, d\mathbf{x} + \rho \int_{\Omega} H_\epsilon(\phi)\, r_{obj}\left(\mathcal{H}(\mathbf{x})\right) + [1 - H_\epsilon(\phi)]\, r_{bg}\left(\mathcal{H}(\mathbf{x})\right)\, d\mathbf{x} \quad (1)$$

where H is the Heaviside, r_{obj} and r_{bg} are *object* and *background* positive monotonically decreasing data-driven functions driven from the grouping criteria. The simplest possible approach would involve the Mumford-Shah approach that aims at separating the means between the two classes. Above equation can be straightforwardly extended in order to deal with other optical or radar data like for example in cases where multi- or hyper-spectral remote sensing data are available.

Furthermore, instead of relying only on the results of an unconstrained evolving surface, we are forcing our output segments to inherit their 2D shape from our prior models. Thus, instead of evolving an arbitrary surface we evolve selected geometric shapes and the 2D prior-based segmentation energy term takes the following form:

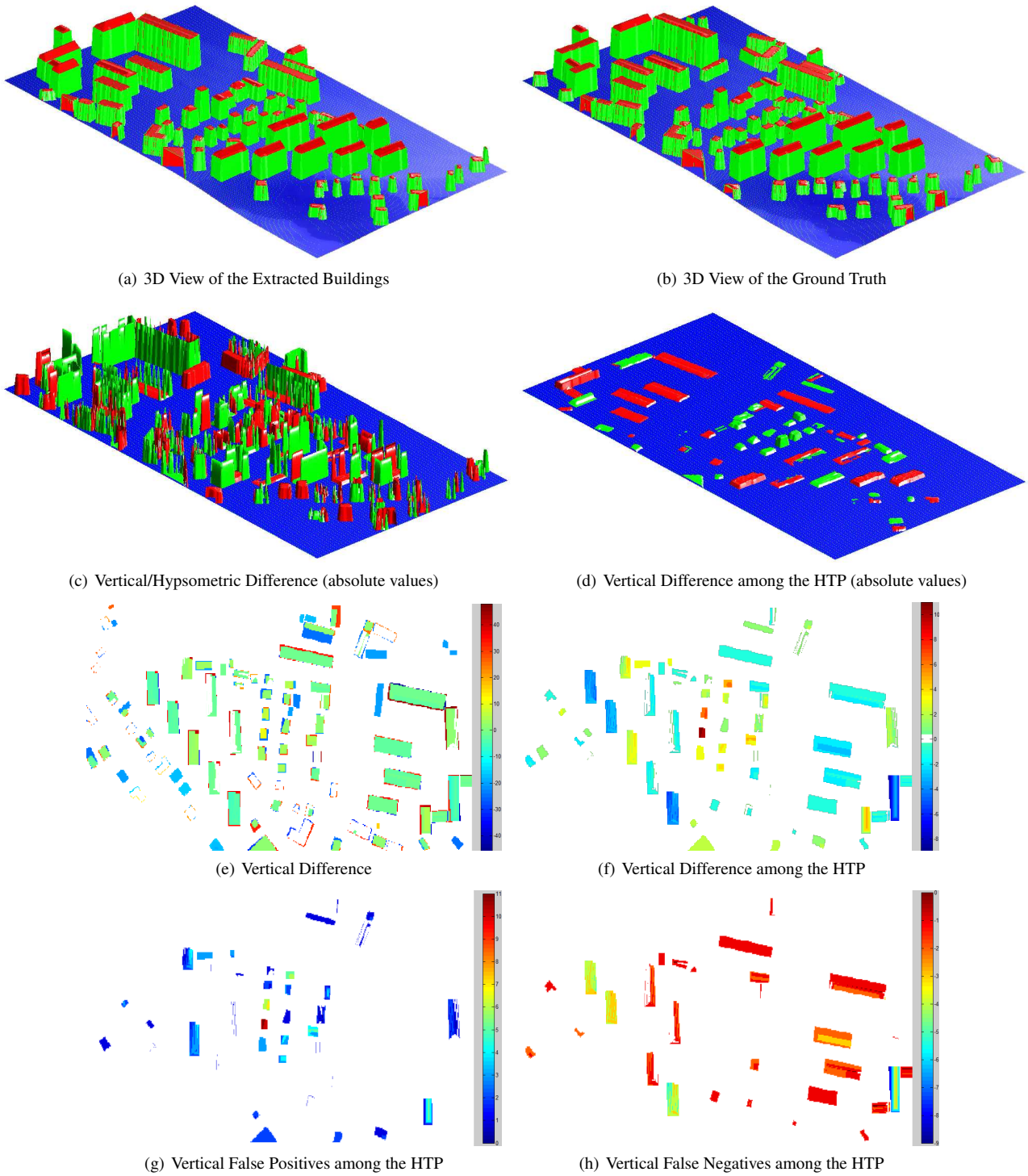
$$E_{2D}(\phi, \mathcal{T}_i, \mathbf{L}) = \sum_{i=1}^{m-1} \int \left(\frac{H_\epsilon(\phi(\mathbf{x})) - H_\epsilon(\tilde{\phi}_i\left(\mathcal{T}_i(\mathbf{x})\right))}{\sigma_i} \right)^2 x_i(\mathbf{L}(\mathbf{x}))d\mathbf{x} + \int \lambda^2 x_m(\mathbf{L}(\mathbf{x}))d\mathbf{x} + \rho \sum_{i=1}^{m} \int |\nabla L(\mathbf{x})|d\mathbf{x} \quad (2)$$

with the two parameters $\lambda, \rho > 0$ and the k-dimensional labeling formulation able for the dynamic labeling of up to $m = 2^k$ regions.

In this way, during optimization the number of selected regions $m = 2^k$ depends on the number of the possible building segments according to ϕ and thus the k-dimensional labeling function $\mathbf{L}$ obtains incrementally multiple instances. It should be, also, mentioned that the initial pose of the priors are not known. Such a formulation $E_{seg} + E_{2D}$ allows data with the higher spatial resolution to constrain properly the footprint detection in order to achieve the optimal spatial accuracy. Furthermore, it solves segmentation simultaneously in both spaces (image and DEM) and addresses fusion in a natural manner.

3.1 Grammar-based Object Reconstruction

In order to determine the 3D geometry of the buildings, one has to estimate the height of the structure with respect to the ground and the orientation angles of the roof components i.e. five unknown parameters: the building's main height h_m which is has a constant value for every building and the four angles ω of the



(a) 3D View of the Extracted Buildings

(b) 3D View of the Ground Truth

(c) Vertical/Hypsometric Difference (absolute values)

(d) Vertical Difference among the HTP (absolute values)

(e) Vertical Difference

(f) Vertical Difference among the HTP

(g) Vertical False Positives among the HTP

(h) Vertical False Negatives among the HTP

Figure 4: Vertical/Hypsometric Difference between the Extracted Buildings and the Ground Truth

roof's inclined planes ($\Theta_i = (h_m, \omega_1, \omega_2, \omega_3, \omega_4)$). These four angles (Figure 2) along with the implicitly derived dimensions of every building's footprint (from E_{2D}) can define the roof's height at every point (pixel) $h_r(x, y)$:

$$\begin{aligned} h_r(x,y) = \\ \min\left[\mathcal{D}(P_1,P_m); \mathcal{D}(P_2,P_m); \mathcal{D}(P_3,P_m); \mathcal{D}(P_4,P_m)\right] \\ = \min\left[d_1 \tan\omega_1;\ d_2 \tan\omega_2;\ d_3 \tan\omega_3;\ d_4 \tan\omega_4\right] \end{aligned} \quad (3)$$

where $\mathcal{D}$: is the perpendicular distance between the horizontal plane P_m and roof's inclined plane $P_{1:4}$. The distance for e.g. between P_1 and P_m in Figure 2 is the actual roof's height at that point (x, y) and can be calculated as the product of the tangent of plane's P_1 angle and the horizontal distance d_1 lying on plane P_m. $\mathcal{D}(P_1, P_m)$ is, also, the minimum distance in that specific point comparing with the ones that are formed with the other three inclined planes.

Utilizing the 3D information from $\mathcal{H}$ -either from point clouds or from a height map- the corresponding energy E_{3D} that recovers

our five unknowns for a certain building i has been formulated as follows:

$$E_{3D}(\Theta_i) = \sum_{i=1}^{m} \int_{\Omega_i} \left(h_{m_i} + h_{r_i}(\mathbf{x}) - \mathcal{H}(\mathbf{x})\right)^2 \, d\mathbf{x} \quad (4)$$

Each prior that has been selected for a specific region is forced to acquire such a geometry so as at every point its total height matches the one from the available DEM. It's a heavily constrained formulation and thus robust. The introduced, here, recognition driven framework now takes the following form in respect to ϕ, $\mathcal{T}_i$, $\mathbf{L}$ and Θ_i:

$$E_{total} \; = \; E_{seg}(\phi) + \mu E_{2D}(\phi, \mathcal{T}_i, \mathbf{L}) + \mu E_{3D}(\Theta_i) \quad (5)$$

The energy term E_{seg} addresses fusion in a natural way and solves segmentation ϕ in both $\mathcal{I}(\mathbf{x})$ and $\mathcal{H}(\mathbf{x})$ spaces. The term E_{2D} estimates which family of priors, i.e which 2D footprint $\mathbf{i}$, under any projective transformation T_i best fit at each segment ($\mathbf{L}$). Finally, the energy E_{3D} recovers the 3D geometry Θ_i of every prior by estimating building's h_m and h_r heights.

4 QUALITATIVE AND QUANTITATIVE ASSESSMENT OF THE PRODUCED 3D MODELS

The quality assessment of 3D data ((Meidow and Schuster, 2005),(Sargent et al., 2007) and their references therein) involves the assessment of both the geometry and topology of the model. During our experiments the quantitative evaluation was performed based on the 3D ground truth data which were derived from a manual digitization procedure. The standard quantitative measures of Completeness (detection rate), Correctness (under-detection rate) and Quality (a normalization between the previous two) were employed. To this end, the quantitative assessment is divided into two parts: Firstly, for the evaluation of the extracted 2D boundaries i.e. the horizontal localization of the building footprints (Figure 3) and secondly, for the evaluation of the hypsometric differences i.e. the vertical differences between the extracted 3D building and the ground truth (Figure 4).

In order to assess the horizontal accuracy of the extracted building footprints the measures of Horizontal True Positives (HTP), Horizontal False Positives (HFP) and Horizontal False Negatives (HFN), were calculated.

$$\begin{aligned} \text{2D Completeness} &= \frac{\text{area of correctly detected segments}}{\text{area of the ground truth}} \\ &= \frac{HTP}{HTP + HFN} \\ \text{2D Correctness} &= \frac{\text{area of correctly detected segments}}{\text{area of all detected segments}} \\ &= \frac{HTP}{HTP + HFP} \\ \text{2D Quality} &= \frac{HTP}{HTP + HFP + HFN} \end{aligned}$$

Moreover, for the evaluation of the hypsometric differences between the extracted buildings and the ground truth the measures of Vertical True Positives (VTP), Vertical False Positives (VFP) and Vertical False Negatives (VFN) were, also, calculated. The VTP are the voxels among, the corresponding Horizontal True Positive pixels, that have the same altitude with the ground truth. Note that Horizontal True Positives may correspond *(i)* to voxels with the same altitude as in the ground truth (VTP) and *(ii)* to voxels with a lower or higher altitude than the ground truth (VFN and VFP, respectively). Thus, the Vertical False Positives are the voxels with an hypsometric difference with the ground truth, containing all the corresponding voxels from the HFP and the corresponding ones from the HTP (those with a higher altitude than the ground truth). Respectively, the Vertical False Negatives are the voxels with an hypsometric difference with the ground truth, containing all the corresponding voxels from the HFN and the corresponding ones from the HTP (those with a lower altitude than the ground truth). To this end, the 3D quantitative assessment was based on the measures of the 3D Completeness (detection rate), 3D Correctness (under-detection rate) and 3D Quality (a normalization between the previous two), which were calculated in the following way:

2D Quantitative Measures		
Completeness	Correctness	Quality
0.84	0.90	0.76

3D Quantitative Measures		
Completeness	Correctness	Quality
0.86	0.86	0.77

Table 1: Pixel- and Voxel-Based Quality Assessment

$$\begin{aligned} \text{3D Completeness} &= \frac{VTP}{VTP + VFN} \\ \text{3D Correctness} &= \frac{VTP}{VTP + VFP} \\ \text{3D Quality} &= \frac{VTP}{VTP + VFP + VFN} \end{aligned}$$

The developed algorithm has been applied to a number of scenes where remote sensing data was available. The algorithm managed in all cases to accurately recover their footprint and overcome low-level misleading information due to shadows, occlusions, etc. In addition, despite the conflicting height similarity between the desired buildings, the surrounding trees and the other objects the developed algorithm managed to robustly recover their 3D geometry as the appropriate priors were chosen (Figure 1). This complex landscape contains a big variety of texture patterns, more than 80 buildings of different types (detached single family houses, industrial buildings, etc) and multiple other objects of various classes. Two aerial images (with a ground resolution of appx. 0.5m) and a the coarser digital surface model (of appx. 1.0m ground resolution) were available. The robustness and functionality of the proposed method is illustrated, also, on Figures 3 and 4, where one can, clearly, observe the Horizontal and the Vertical True Positives, respectively. The proposed generic variational framework managed to accurately extract the 3D geometry of scene's buildings, searching among various footprint shapes and various roof types. The performed quantitative evaluation reported an overall horizontal detection correctness of 90% and an overall horizontal detection completeness of 84% (Table 1).

In Figure 4c, the hypsometric/vertical difference between the extracted buildings and the ground truth is shown. With a red color are the VFN voxels and with a green color the VFP ones. Similarly, at Figure 4c where the -corresponding among the HTP pixels- VFN and VFP voxels are shown. The performed quantitative evaluation reported an overall 3D completeness and correctness of appx. 86% (Table 1).

5 CONCLUSIONS AND FUTURE WORK

We have developed a generalized variational framework which addresses large-scale reconstruction through information fusion and competing grammar-based 3D priors. We have argued that our inferential approach significantly extends previous 3D extraction and reconstruction efforts by accounting for shadows, occlusions and other unfavorable conditions and by effectively narrowing the space of solutions due to our novel grammar representation and energy formulation. The successful recognition-driven results along with the reliable estimation of buildings 3D geometry suggest that the proposed method constitutes a highly promising tool for various object extraction and reconstruction tasks.

Our a framework can be easily extended to process spectral information, by formulating respectively the region descriptors and to account for other types of buildings or other terrain features. For real-time applications, the labeling function straightforwardly allows a parallel computing formulation by concurrently recovering the transformations for every region. In order to address the sub-optimality of the obtained solution, the use of the compressed sensing framework by collecting a comparably small number of measurements rather than all pixel values is currently under investigation. Last, but not least introducing hierarchical procedural grammars can reduce the complexity of the prior model and provide access to more efficient means of optimization.

ACKNOWLEDGEMENTS

This work has been partially supported from the Conseil General de Hauts-de-Seine and the Region Ile-de-France under the TERRA NUMERICA grant of the Pole de competitivite CapDigital.

REFERENCES

Baltsavias, E., 2004. Object extraction and revision by image analysis using existing geodata and knowledge: current status and steps towards operational systems. ISPRS Journal of Photogrammetry and Remote Sensing 58, pp. 129–151.

Brenner, C., 2005. Building reconstruction from images and laser scanning. International Journal of Applied Earth Observation and Geoinformation 6, pp. 187–198.

Dick, A. R., Torr, P. H. S. and Cipolla, R., 2004. Modelling and interpretation of architecture from several images. International Journal of Computer Vision 60(2), pp. 111–134.

Drauschke, M., Schuster, H.-F. and Förstner, W., 2006. Detectibility of buildings in aerial images over scale space. In: ISPRS Symposium of Photogrammetric Computer Vision, Vol. XXXVInumber Part 3, pp. 7–12.

Forlani, G., Nardinocchi, C., Scaioni, M. and Zingaretti, P., 2006. Complete classification of raw LIDAR data and 3D reconstruction of buildings. Pattern Anal. Appl. 8(4), pp. 357–374.

Hu, J., You, S. and Neumann, U., 2003. Approaches to large-scale urban modeling. IEEE Computer Graphics and Applications 23(6), pp. 62–69.

Jaynes, C., Riseman, E. and Hanson, A., 2003. Recognition and reconstruction of buildings from multiple aerial images. Computer Vision and Image Understanding 90(1), pp. 68–98.

Karantzalos, K. and Paragios, N., 2009. Recognition-Driven 2D Competing Priors Towards Automatic and Accurate Building Detection. IEEE Transactions on Geoscience and Remote Sensing 47(1), pp. 133–144.

Kim, Z. and Nevatia, R., 2004. Automatic description of complex buildings from multiple images. Computer Vision and Image Understanding 96(1), pp. 60–95.

Lafarge, F., Descombes, X., Zerubia, J. and Pierrot-Deseilligny, M., 2007. 3D city modeling based on hidden markov model. In: Proc. IEEE International Conference on Image Processing (ICIP), Vol. II, pp. 521–524.

Matei, B.C.and Sawhney, H., Samarasekera, S., Kim, J. and Kumar, R., 2008. Building segmentation for densely built urban regions using aerial lidar data. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–8.

Mayer, H., 2008. Object extraction in photogrammetric computer vision. ISPRS Journal of Photogrammetry and Remote Sensing 63(2), pp. 213–222.

Meidow, J. and Schuster, H., 2005. Voxel-based quality evaluation of photogrammetric building acquisitions. In: ISPRS International archives of photogrammetry, remote sensing and spatial information sciences (Stilla U, Rottensteiner F, Hinz S (Eds)), Vol. XXXVI, Part 3/W24.

Muller, P., Wonka, P., Haegler, S., Ulmer, A. and Gool, L., 2006. Procedural modeling of buildings. Proceedings of ACM SIGGRAPH / ACM Transactions on Graphics 25(3), pp. 614–623.

Muller, P., Zeng, G., Wonka, P. and Gool, L., 2007. Image-based procedural modeling of facades. Proceedings of ACM SIGGRAPH 2007 / ACM Transactions on Graphics.

Paparoditis, N., Souchon, J.-P., Martinoty, G. and Pierrot-Deseilligny, M., 2006. High-end aerial digital cameras and their impact on the automation and quality of the production workflow. ISPRS Journal of Photogrammetry and Remote Sensing 60(6), pp. 400–412.

Paragios, N., Chen, Y. and Faugeras, O., 2005. Handbook of Mathematical Models of Computer Vision. Springer.

Rottensteiner, F., Trinder, J., Clode, S. and Kubik, K., 2007. Building detection by fusion of airborne laser scanner data and multi-spectral images: Performance evaluation and sensitivity analysis. ISPRS Journal of Photogrammetry and Remote Sensing 62(2), pp. 135–149.

Sargent, I., Harding, J. and Freeman, M., 2007. Data Quality in 3D: Gauging Quality Measures From Users' Requirements. In: International Symposium on Spatial Quality, Endchede, Netherlands.

Sohn, G. and Dowman, I., 2007. Data fusion of high-resolution satellite imagery and LiDAR data for automatic building extraction. ISPRS Journal of Photogrammetry and Remote Sensing 62(1), pp. 43–63.

Suveg, I. and Vosselman, G., 2004. Reconstruction of 3D building models from aerial images and maps. ISPRS Journal of Photogrammetry and Remote Sensing 58, pp. 202–224.

Verma, V., Kumar, R. and Hsu, S., 2006. 3D building detection and modeling from aerial lidar data. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 2213–2220.

Wilczkowiak, M., Sturm, P. and Boyer, E., 2005. Using geometric constraints through parallelepipeds for calibration and 3D modeling. IEEE Transactions on Pattern Analysis and Machine Intelligence 27(2), pp. 194–207.

Zebedin, L., Bauer, J., Karner, K. and Bischof, H., 2008. Fusion of feature-and area- based information for urban buildings modeling from aerial imagery. In: European Conference on Computer Vision, Vol. 5305, Lecture Notes in Computer Science, pp. 873–886.

Zhu, Z. and Kanade (Eds.), T., July, 2008. Special Issue: Modeling and Representations of Large-Scale 3D scenes. International Journal of Computer Vision.

OBJECT EXTRACTION FROM LIDAR DATA USING AN ARTIFICIAL SWARM BEE COLONY CLUSTERING ALGORITHM

S. Saeedi [a], F. Samadzadegan [b], N. El-Sheimy [a]

[a] Department of Geomatics Engineering, University of Calgary, T2N 1N4, AB, Canada - (ssaeedi, elsheimy)@ucalgary.ca
[b] Department of Geomatics Engineering, Faculty of Engineering, University of Tehran, Iran - samadz@ut.ac.ir

Commission III, WG III/4

KEY WORDS: Clustering, LIDAR, Artificial Bee Colony, Urban Area, Object Extraction

ABSTRACT:

Light Detection and Ranging (LIDAR) systems have become a standard data collection technology for capturing object surface information and 3D modeling of urban areas. Although, LIDAR systems provide detailed valuable geometric information, they still require extensive interpretation of their data for object extraction and recognition to make it practical for mapping purposes. A fundamental step in the transformation of the LIDAR data into objects is the segmentation of LIDAR data through a clustering process. Nevertheless, due to scene complexity and the variety of objects in urban areas, e.g. buildings, roads, and trees, clustering using only one single cue will not reach meaningful results. The multi dimensionality nature of LIDAR data, e.g. laser range and intensity information in both first and last echo, allow the use of more information in the data clustering process and ultimately into the reconstruction scheme. Multi dimensionality nature of LIDAR data with a dense sampling interval in urban applications, provide a huge amount of valuable information. However, this amount of information produces a lot of problems for traditional clustering techniques. This paper describes the potential of an artificial swarm bee colony optimization algorithm to find global solutions to the clustering problem of multi dimensional LIDAR data in urban areas. The artificial bee colony algorithm performs neighborhood search combined with random search in a way that is reminiscent of the food foraging behavior of swarms of honey bees. Hence, by integrating the simplicity of the *k*-means algorithm with the capability of the artificial bee colony algorithm, a robust and efficient clustering method for object extraction from LIDAR data is presented in this paper. This algorithm successfully applied to different LIDAR data sets in different urban areas with different size and complexities.

1. INTRODUCTION

The need for rapidly generating high-density digital elevation data for areas of considerable spatial extent has been one of the main motives for the development of commercial airborne laser scanning systems. During the last decade, several clustering and filtering techniques have been developed for the extraction of 3D objects for city modelling applications or removing the "artefacts" of bare terrain (i.e. Buildings and trees) in order to obtain the true Digital Elevation Model (Filin and Pfeifer; 2006; Kraus and Pfeifer, 1998; Lodha et al., 2007; Rottensteiner and Briese, 2002; Tóvári and Vögtle, 2004).

However due to low information content and resolution of available commercial LIDAR scanners, it is difficult to correctly recognize and remove 3D objects exclusively from LIDAR range data in urban areas (Maas, 2001; Samadzadegan, 2004; Tao and Hu, 2001; Vosselman et al., 2004).

In order to improve the performance of 3D object extraction process, additional data should be considered. Most LIDAR systems register, at least, two echoes of the laser beam, the first and the last echo, which generally correspond to the highest and the lowest object point hit by the laser beam. First and last echo data will especially differ in the presence of vegetation (Kraus, 2002). Moreover, LIDAR systems record the intensity of the returned laser beam which is mainly in the infrared part of the electromagnetic spectrum. In addition, an extra powerful source of information is visible image. Digital images can provide additional information through their intensity and spectral content as well as their high spatial resolution which is better than the resolution of laser scanner data.

Therefore, in the context of 3D object extraction in urban areas, various type of information can be fused to overcome the difficulties of classification and identification of complicated objects (Lim and Suter, 2007; Vosselman et al., 2004). Collecting this information, extremely enlarge the size of data sets and proportionally the dimension of feature spaces in clustering process. As a result, most of traditional clustering techniques that have been applied with standard data and low feature space dimension are not efficient enough for object extraction process from LIDAR data (Melzer, 2007; Lodha et al., 2007).

k-means is one of the most popular clustering algorithms for handling massive datasets. The algorithm is efficient at clustering large data sets because its computational complexity only grows linearly with the number of data points (Kotsiantis and Pintelas, 2004). However, the algorithm may converge to solutions that are not optimal. This paper presents an artificial bee colony (ABC) clustering algorithm for overcoming the existing problems of traditional *k*-means.

2. BASIC CONCEPTS IN DATA CLUSTERING

Historically, the notion of finding useful patterns in data has been given a variety of names including data clustering, data mining, knowledge discovery, pattern recognition, information extraction, etc (Ajith et al., 2006). Data clustering is an analytic process designed to explore data by discovering of consistent patterns and/or systematic relationships between variables, and then to validate the findings by applying the detected patterns to new subsets of data.

Data clustering is a difficult problem in unsupervised pattern recognition as the clusters in data may have different shapes and sizes. In the background of clustering techniques, the following terms are used in this paper (Jain et al., 1999):

- A pattern (or feature vector), z, is a single object or data point used by the clustering algorithm.
- A feature (attribute) is an individual component of a pattern.
- A cluster is a set of similar patterns, and patterns from different clusters are not similar.
- A distance measure is a metric used to evaluate the similarity of patterns.

The clustering problem can be formally defined as follows (Jain et al., 1999): Given a data set $Z=\{z_1,z_2, \ldots ,z_p, \ldots ,z_{Np}\}$ where z_p is a pattern in the N_d-dimensional feature space, and N_p is the number of patterns in Z, then the clustering of Z is the partitioning of Z into K clusters $\{C_1,C_2, \ldots ,C_K\}$ satisfying the following conditions:

- Each pattern should be assigned to a cluster, i.e.
$$\cup_{j=1}^{k} C_j = Z$$
- Each cluster has at least one pattern assigned to it, i.e.
$$C_k \neq 0, \qquad k = 1, \ldots, K$$
- Each pattern is assigned to one and only one cluster
$$C_k \cap C_j = 0, where\ k \neq j$$

As previously mentioned, clustering is the process of identifying natural groupings or clusters within multidimensional data based on feature space through similarity measure. Hence, similarity measures are fundamental components in most clustering algorithms (Jain et al., 1999). The most popular way to evaluate a similarity measure is the use of distance measures. The most widely used distance measure is the Euclidean distance, defined as:

$$d(z_i, z_j) = \sqrt{\sum_{k=1}^{N_d} (z_{i,k} - z_{j,k})^2} = \|z_i - z_j\|_2 \qquad (1)$$

Generally, clustering algorithms can be categorized into partitioning methods, hierarchical methods, density-based methods, grid-based methods, and model-based methods. An excellent survey of clustering techniques can be found in (Kotsiantis and Pintelas, 2004). In this paper, the focus will be on the partitional clustering algorithms. Partitional clustering algorithms divide the data set into a specified number of clusters and then evaluate them by some criteria. These algorithms try to minimize certain criteria (e.g. a square error function) and can therefore be treated as optimization problems (Harvey et al., 2002; Omran et al., 2005; Wilson et al., 2002).

The most widely used partitional algorithm in clustering techniques is the iterative *k*-means approach (Kotsiantis and Pintelas, 2004). The objective function J that the *k*-means optimizes is:

$$J_{K-means} = \sum_{j=1}^{K} \sum_{\forall z_p \in C_k} d^2 (z_p, m_k) \qquad (2)$$

Where m_k is the centroid of the *k*-th cluster. The membership and weight functions u for *k*-means are defined as:

$$u(m_k|z_p) = \begin{cases} 1 & if\ d^2(z_p, m_k) = \arg min_k \{d^2(z_p, m_k)\} \\ 0 & otherwise \end{cases} \qquad (3)$$

Consequently, the k-means method minimizes the intra-cluster distance. The *k*-means algorithm starts with *k* centroids (initial values are randomly selected or derived from a priori information). Then, each pattern z_p in the data set is assigned to the closest cluster (i.e. closest centroid). Finally, the centroids are recalculated according to the associated patterns. This procedure is repeated until convergence is achieved.

It is known that the *k*-means algorithm may reach local optimal solutions, depending on the choice of the initial cluster centres. Genetic algorithms have a potentially greater ability to avoid local optima through the localised search employed by most clustering techniques. Maulik and Bandyopadhyay (2004) proposed a genetic algorithm-based clustering technique, called GA-clustering, that proven to be effective in optimal clusters. With this algorithm, solutions (typically, cluster centroids) are represented by bit strings. The search for an appropriate solution begins with a population, or collection, of initial solutions. Members of the current population are used to create the next generation population by applying operations such as random mutation and crossover. At each step, the solutions in the current population are evaluated relative to some measures of fitness (which, typically, is inversely proportional to *d*), with the fittest solutions selected probabilistically as seeds for producing the next generation. The process performs a generate-and-test beam search of the solution space, in which variants of the best current solutions are most likely to be considered next. In the next section, an alternative clustering method to solve the local optimum problem of the *k*-means algorithm is described. The applied method adopts the artificial swarm bees algorithm as it has proved to give a more robust performance than other intelligent optimisation methods for a range of complex problems (Pham, 2006).

3. CLUSTERING OF LIDAR DATA USING SWARM ARTIFICIAL BEE COLONY ALGORITHM

Swarm Intelligence (SI) is an innovative distributed intelligent paradigm for solving optimization problems that originally took its inspiration from the biological examples by swarming, flocking and herding phenomena. These techniques incorporate swarming behaviours observed in flocks of birds, schools of fish, or swarms of bees, and even human social behaviour, from which the idea is emerged (Omran et al., 2002, 2005; Paterlini and Krink, 2005; Pham et al., 2006; Wu and Shi, 2001). Data clustering and swarm intelligence may seem that they do not have many properties in common. However, recent studies suggest that they can be used together for several real world data clustering and mining problems especially when other methods would be too expensive or difficult to implement.

Clustering approaches inspired by the collective behaviours of ants have been proposed by Wu and Shi (2001), Labroche et al. (2001). The main idea of these approaches is that artificial ants are used to pick up items and drop them near similar items resulting in the formation of clusters. Omran et al. (2002) proposed particle swarm optimization (PSO) clustering algorithm. The results of Omran et al. (2002, 2005) show that PSO outperformed *k*-means, fuzzy *c*-means (FCM) and other state-of-the-art clustering algorithms. More recently, Paterlini and Krink (2005) compared the performance of *k*-means, genetic algorithm (GA), PSO and Differential Evolution (DE) for a representative point evaluation approach to partitional clustering. The results show that GAs, PSO and DE outperformed the *k*-means algorithm. Pham et al. (2006) used the artificial bee colony algorithm for clustering of different data sets. The obtained results of their work show that their proposed artificial bee colony algorithm has better performance than both of standard *k*-means as well as GA-based method. In general, the literature review of recent

techniques in clustering shows that the swarm-based clustering algorithm performs better than the *k*-means algorithm. Clustering of massive LIDAR data and the unique potential of artificial bee colony algorithm in solving complex optimization problems are the core of this paper. The research work presented in this paper clearly show that the artificial swarm bee colony algorithm has clearly outperform *k*-means method in clustering of LIDAR data.

3.1 Artificial Bee Colony Algorithm

A colony of honey bees can extend itself over long distances in order to exploit a large number of food sources (Camazine et al., 2003; Pham et al., 2006) . The foraging process begins in a colony by scout bees being sent to search for promising flower patches. Flower patches with large amounts of nectar or pollen that can be collected with less effort tend to be visited by more bees, whereas patches with less nectar or pollen receive fewer bees (Camazine et al., 2003).

In the artificial bee algorithms, a food source position represents a possible solution to the problem to be optimized. Therefore, at the initialization step, a set of food source positions are randomly produced and also the values of control parameters of the algorithm are assigned. The nectar mount of a food source corresponds to the quality of the solution represented by that source. So the nectar amounts of the food sources existing at the initial positions are determined. In other words, the quality values of the initial solutions are calculated.

Each employed bee is moved onto her food source area for determining a new food source within the neighbourhood of the present one, and then its nectar amount is evaluated. If the nectar amount of the new one is higher, then the bee forgets the previous one and memorizes the new one. After the employed bees complete their search, they come back into the hive and share their information about the nectar amounts of their sources with the onlookers waiting on the dance area. All onlookers successively determine a food source area with a probability based on their nectar amounts. If the nectar amount of a food source is much higher when compared with other food sources, it means that this source will be chosen by most of the onlookers. This process is similar to the natural selection process in evolutionary algorithms. Each onlooker determines a neighbour food source within the neighbourhood of the one to which she has been assigned and then its nectar amount is evaluated.

3.2 Artificial Swarm Bee Colony Clustering Method

The artificial swarm bee colony clustering method exploits the search capability of the Bees Algorithm to overcome the local optimum problem of the *k*-means algorithm. More specifically, the task is to search for appropriate cluster centres (c_1, c_2,...,c_k) such that the clustering metric *d* (equation 1) is minimised. The basic steps of this clustering operation are:

1. Initialise the solution population.
2. Evaluate the fitness of the population.
3. While (stopping criterion is not met)
   a. Form new population.
   b. Select sites for neighbourhood search by means of information in the neighbourhood of the present one.
   c. Recruit bees for selected sites (more bees for the best e sites) and evaluate fitness values.
   d. Select the fittest bee from each site.
   e. Assign remaining bees to search randomly and evaluate their fitness values.

End While.

Each bee represents a potential clustering solution as set of *k* cluster centres and each site represent the patterns or data objects. The algorithm requires some parameters to be set, namely: number of scout bees (*n*), number of sites selected for neighbourhood searching (*m*), number of top-rated (*elite*) sites among *m* selected sites (*e*), number of bees recruited for the best *e* sites (*nep*), number of bees recruited for the other (*me*) selected sites (*nsp*), and the stopping criterion for the loop.

At the initialization stage, a set of scout bee population (*n*) are randomly selected to define the *k* clusters. The Euclidean distances between each data pattern and all centres are calculated to determine the cluster to which the data pattern belongs. In this way, initial clusters can be constructed. After the clusters have been formed, the original cluster centres are replaced by the actual centroids of the clusters to define a particular clustering solution (i.e. a bee). This initialization process is applied each time new bees are to be created.

In step 2, the fitness computation process is carried out for each site visited by a bee by calculating the clustering metric *d* (equation 1) which is inversely related to fitness. Step 3, is the main step of bee colony optimization, which start by forming new population (step 3-a). In step 3-b, the *m* sites with the highest fitness are designated as "selected sites" and chosen for neighbourhood search. In steps 3-c and 3-d, the algorithm conducts searches around the selected sites, assigning more bees to search in the vicinity of the best *e* sites. Selection of the best sites can be made directly according to the fitness associated with them. Alternatively, the fitness values are used to determine the probability of the sites being selected. Searches in the neighbourhood of the best e sites - those which represent the most promising solutions - are made more detailed. As already mentioned, this is done by recruiting more bees for the best *e* sites than for the other selected sites. Together with scouting, this differential recruitment is a key operation of the bee algorithm. In step 3-d, only the bee that has found the site with the highest fitness (the "fittest" bee) will be selected to form part of the next bee population. In nature, there is no such a restriction. This restriction is introduced here to reduce the number of points to be explored. In step 3-e, the remaining bees in the population are assigned randomly around the search space to scout for new potential solutions. At the end of each loop, the colony will have two stages to its new population: representatives from the selected sites, and scout bees assigned to conduct random searches. These steps are repeated until a stopping criterion is met.

4. EXPERIMENTAL INVESTIGATIONS

The airborne LIDAR data used in the experimental investigations have been recorded with TopScan's Airborne Laser Terrain Mapper system ALTM 1225 (TopScan, 2004). The data are recorded in area of Rheine in Germany. Two different patches with residential and industrial pattern were selected to test the developed algorithm. The selected areas were suitable for the evaluation of the proposed classification strategy because the required complexities (e.g. proximities of different objects e.g. building and tree) were available in the image (figure 1-a, b). The pixel size of the range images is one meter. This reflects the average density of the irregularly recorded 3D points which is fairly close to one point per m^2. Intensity images for the first and last echo data have been also recorded and the intention was to use them in the experimental investigations, Figure 1 shows the details of the test data. The impact of trees in the first and last echo images can be easily recognized by comparing the two images of this figure.

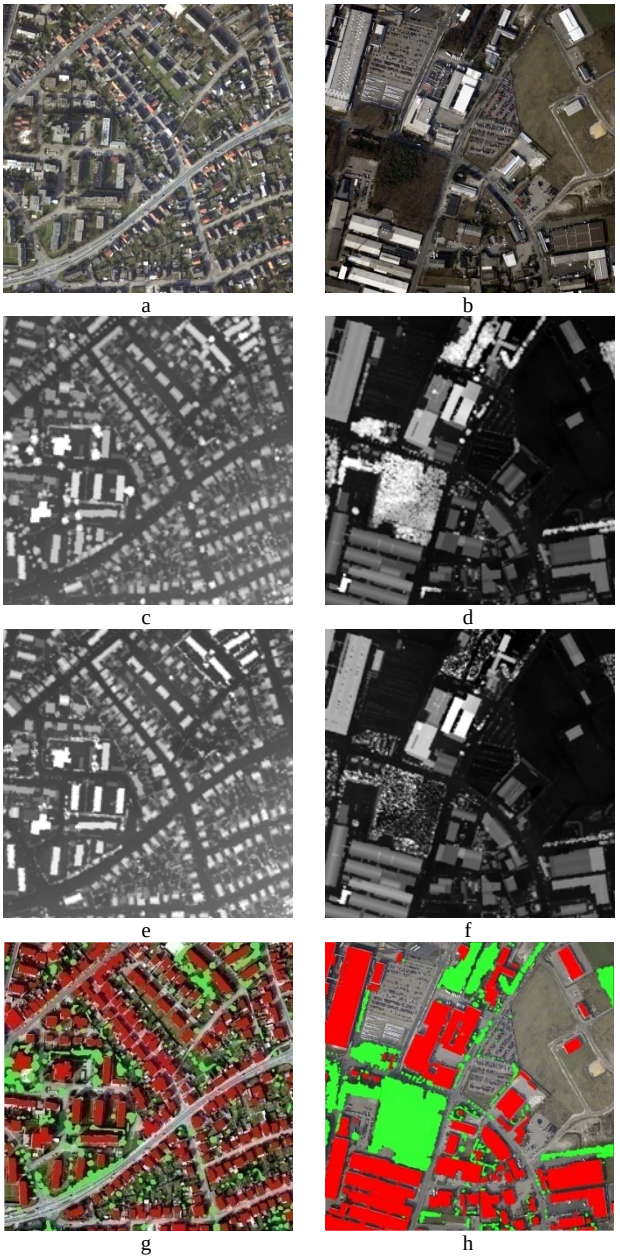


Figure 1. a) Aerial image of residential area. b) Aerial image of industrial area. c) First echo LIDAR range data of residential area. d) First echo LIDAR range data of industrial area. e) Last echo LIDAR range data of residential area. f) Last echo LIDAR range data of industrial area. g) Overlaid of manually digitized objects in residential area; h) Overlaid of manually digitized objects in residential area

The first step in every clustering process is to extract the feature image bands. The features of theses feature bands should carry useful textural or surface related information to differentiate between regions related to the surface. Several features have been proposed for clustering of range data. Axelsson (1999) employs the second derivatives to find textural variations and Maas (1999) utilizes a feature vector including the original height data, the Laplace operator, maximum slope measures and others in order to classify the data. In the following experiments we used five types of features:

- LIDAR range data
- The difference between first and last echo range images
- Top-Hat filtered last echo range image

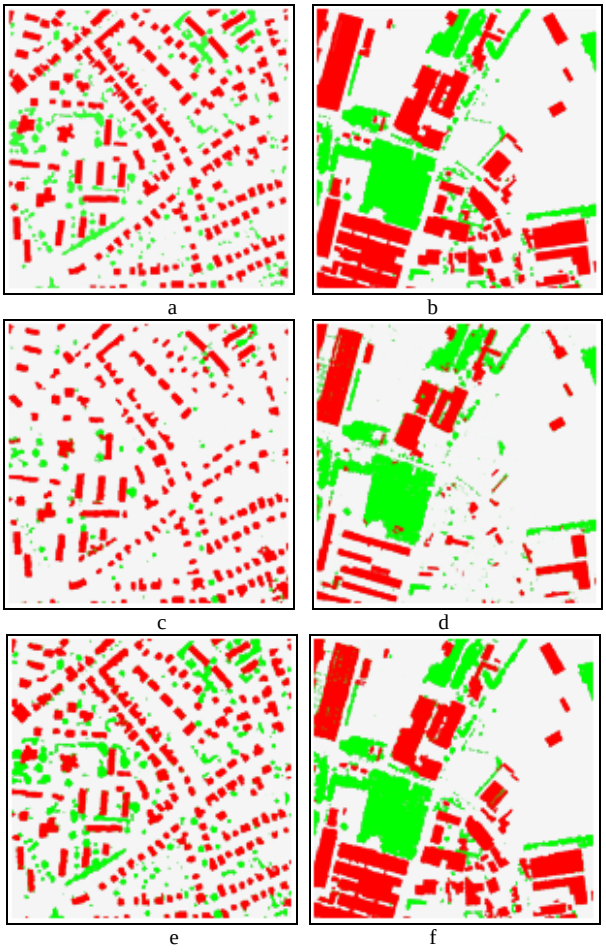


Figure 2. a) Manually digitized objects in residential area. b) Manually digitized objects in industrial area. c) Clustering results of *k*-means in residential area. d) Clustering results of *k*-means in industrial area. e) Clustering results of artificial swarm bee colony algorithm in residential area. f) Clustering results of swarm bee algorithm in industrial area.

- Local height variation which is computed using a small window (3*3) around a data sample.
- Last echo intensity

The normalized difference of the first and last echo range images is used as the major feature band for discrimination of the vegetation pixels from the others. According to the above defined features, the *k*-means and artificial swarm bee algorithm were developed based on the parameters listed in table 1.

Table 1. Parameters used in the clustering of LIDAR datasets

Algorithm	Parameters Value	
***k*-means**	Maximum number of iterations	1000
Artificial swarm bee colony algorithm	Number of scout bees, *n*	35
	Number of sites selected for neighbourhood search, *m*	11
	Number of best "elite" sites out of *m* selected sites, *e*	2
	Number of bees recruited for best *e* sites, *nep*	7
	Number of bees recruited for the other (m-e) selected sites, *nsp*	3
	Number of iterations, *R*	200

Evaluation of these two algorithms for clustering of the data sets into three clusters (ground, tree, and building) is depicted in figure 2. Figures 2c and 2d show the *k*-means clustering results and figures 2e and 2f show the artificial bee colony algorithm clustering results in two evaluation areas. Building class regions are highlighted by red and vegetation class regions by green colour in figure 2. Visual inspections shows that vegetation class is directly associated with trees, bushes or forest and the building class is mainly associated with building regions.

4.1 Accuracy Assessment

Comparative studies on clustering algorithms are difficult due to lack of universally agreed upon quantitative performance evaluation measures. Many similar works in clustering use the classification error as the final quality measurement (Zhong and Ghosh, 2003); so in this research, we adopt a similar approach. In this paper, confusion matrix used to evaluate the true labels and the labels returned by the clustering algorithms as the quality assessment measure. If some ground truth is available, the relation between the "true" classes and the classification result can be quantified. With the clusters the same principle can be applied. Mostly a much higher number of clusters is then related to the given ground truth classes to examine the quality of the clustering algorithm. From the confusion matrix we calculate the *Kappa Coefficient* (Cohen, 1960). Although the accuracy measurements described above, namely, the overall accuracy, producer's accuracy, and user's accuracy, are quite simple to use, they are based on either the principal diagonal, columns, or rows of the confusion matrix only, which does not use the complete information from the confusion matrix. A multivariate index called the Kappa coefficient (Tso and Mather, 2009) overcomes these limitations. The Kappa coefficient uses all of the information in the confusion matrix in order for the chance allocation of labels to be taken into consideration. The Kappa coefficient is defined by:

$$\hat{k} = \frac{N\sum_{i=1}^{r} x_{ii} - \sum_{i=1}^{r}(x_{i+} \times x_{+i})}{N^2 - \sum_{i=1}^{r}(x_{i+} \times x_{+i})} \qquad (4)$$

In this equation, $\hat{k}$ is the kappa coefficient, r is the number of columns (and rows) in a confusion matrix, x_{ii} is entry *(i, i)* of the confusion matrix, x_{i+} and x_{+i} are the marginal totals of row i and column j, respectively, and N is the total number of observations (Tso and Mather, 2009).

Table 2 shows the confusion matrix and Kappa coefficient of *k*-means and artificial swarm bee colony algorithms clustering in residential dataset. The confusion matrix and Kappa coefficient of *k*-means and artificial swarm bee colony algorithms clustering in industrial dataset presented in Table 3.

By comparing the counts in each class, a striking difference to the artificial swarm bee colony algorithm result is clearly observed. For the two classes of major interest in this study, the building class and tree class, the differences are quite significant. Visual interpretation clearly indicates that the building class of k-means not only include building areas but also regions related to roads which supports the smaller number of counts of the artificial swarm bee colony method to be more precise. Similarly the higher number of counts for the tree class indication (3D) vegetation regions (trees, bushes) obtained with the artificial swarm bee colony algorithm method is supported by visual interpretation. Overall performance of artificial bee colony algorithm is outperforming k-means clustering algorithm. This can be observed from the Kapa coefficient.

Table 2. Confusion matrix and Kappa coefficient of *k*-means and artificial swarm bee colony algorithms in residential area.

k-means	Reference Data			
	Building	Tree	Ground	Total
Building	64338	1551	338	66227
Tree	3561	58692	5930	68183
Ground	54341	10509	290740	355590
Total	122240	70752	297008	490000
Kappa coefficient = 0.6927				

Bee algorithms	Reference Data			
	Building	Tree	Ground	Total
Building	114602	3471	5686	123759
Tree	2124	61123	6144	69391
Ground	4214	7558	285078	296850
Total	120940	72152	296908	490000
Kappa coefficient = 0.8916				

Table 3. Confusion matrix and Kappa coefficient of *k*-means and artificial swarm bee colony algorithms in industrial area.

k-means	Reference Data			
	Building	Tree	Ground	Total
Building	26878	2168	1108	30154
Tree	187	3707	105	3999
Ground	16443	12879	139025	168347
Total	43508	18754	140238	202500
Kappa coefficient = 0.584				

Bee algorithms	Reference Data			
	Building	Tree	Ground	Total
Building	39528	1158	2097	42783
Tree	839	15641	1290	17770
Ground	3842	3483	134622	141947
Total	44209	20282	138009	202500
Kappa coefficient = 0.866				

5. CONCLUSION

This paper presented and tested a new clustering method based on the artificial bee colony algorithm in extracting buildings and trees form LIDAR data. The method employs the artificial swarm bee colony algorithm to search for the set of cluster centres that minimizes a given clustering metric. One of the advantages of this method is that it does not become trapped at locally optimal solutions. This is due to the ability of the artificial swarm bee colony algorithm to perform local and global search simultaneously. Experimental results for different LIDAR data sets have demonstrated that the artificial swarm bee colony algorithm method produces better performances than those of the *k*-means algorithm. One of the drawbacks of the artificial artificial swarm bee colony algorithm, however, is the number of tunable parameters it employs.

6. ACKNOWLDGMNT

The authors would like to thank Dr. Michael Hahn from Stuttgart University of Applied Sciences for providing the data set used in the paper.

7. REFERENCES

Ajith, A., Crina G., Vitorino R., (Eds.) (2006). Swarm Intelligence in Data Mining, Studies in Computational Intelligence (series), Vol. 34, Springer-Verlag, ISBN: 3-540-34955-3, 267 p., Hardcover.

Axelsson, 1999 P. Axelsson (1999), Processing of laser scanner data-algorithms and applications, ISPRS Journal of Photogrammetry and RS 54 (2–3), pp. 138–147. Article | PDF (1269 K) | View Record in Scopus | Cited By in Scopus (113)

Camazine, S., Deneubourg, J., Franks, N.R., Sneyd, J., Theraula, G. and Bonabeau, E. (2003). Self-Organization in Biological Systems, (Princeton University Press, Princeton).

Filin, S. and Pfeifer, N., (2006). Segmentation of airborne laser scanning data using a slope adaptive neighborhood. *ISPRS Journal of Photogrammetry and Remote Sensing* 60(2): 71-80.

Harvey N. R., J. Theiler, S. P. Brumby, S. Perkins, J. J. Szymanski, J. J. Bloch, R. B. porter, M. Galassi, A. C. Young, (2002). "Comparison of GENIE and conventional supervised classifiers for multispectral image feature extraction," IEEE Trans. on Geoscience and RS, vol. 40, no. 2, pp. 393-404.

Jain, A.K. , M.N. Murty, P.J. Flynn, (1999). Data Clustering: A Review, ACM Computing Surveys, Vol. 31, No. 3, Sep. 1999.

Kotsiantis S. and Pintelas P., (2004). Recent advances in clustering: a brief survey, WSEAS Transactions on Information Science and Applications 1:73-81.

Kraus and Pfeifer, (1998). Determination of terrain models in wooded areas with ALS data. *ISPRS Journal of Photogrammetry and Remote Sensing* **53** 4 (1998), pp. 193–20.

Kraus, K., (2002). Principles of airborne laser scanning. Journal of the Swedish Society for Photogrammetry and Remote Sensing 2002:1 53–56.

Labroche N, Monmarche N, and Venturini G., (2002). Visual Clustering based on the Chemical Recognition System on Ants. In Proceedings of the European Conf. on AI, 2002.

Lim, E. H. and Suter, D., (2007). Conditional Random Field for 3D Point Clouds with Adaptive Data Reduction. In Proceedings of the 2007 international Conference on Cyberworlds (October 24 - 26, 2007). International Conference on Cyberworlds. IEEE Computer Society, Washington, DC, 404-408. DOI= http://dx.doi.org/10.1109/CW.2007.24

Lodha, S. K., Fitzpatrick, D. M., and Helmbold, D. P., (2007). Aerial Lidar Data Classification using AdaBoost. In Proceedings of the Sixth international Conference on 3-D Digital Imaging and Modeling (August 21 - 23, 2007). 3DIM. IEEE Computer Society, Washington, DC, 435-442. DOI= http://dx.doi.org/10.1109/3DIM.2007.10

Maas, H.-G., (1999). Close solutions for the determination of parametric house models from invariant moments of airborne laserscanner data, *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 32(3-2W5): 193-199.

Maas, H.-G., (2001). The suitability of Airborne Laser Scanner Data for Automatic 3D Object Reconstruction, Third Int. Workshop on Automatic Extraction of Man-Made Objects from Aerial and Space Images, Ascona, Switzerland.

Maulik U. and Bandyopadhyay S., (2004). "Genetic Algorithm Based Clustering Technique", Pattern Recognition, vol. 33, no. 9, pp. 1455-1465, 2004.

Melzer T., (2007); "Non-parametric segmentation of ALS point clouds using mean shift"; Journal of Applied Geodesy. Volume 1, Issue 3, Pages 159–170, ISSN (Online) 1862-9024, ISSN (Print) 1862-9016, DOI: 10.1515/jag.2007.018

Omran M, Salman A, and Engelbrecht AP., (2002). Image Classification using Particle Swarm Optimization. In Conf. on Simulated Evolution and Learning, vol. 1, pp 370–374, 2002.

Omran M., A. Engelbrecht and A. Salman, (2005). Differential evolution methods for unsupervised image classification, In Proc. of the IEEE Cong. on Evolutionary Computation (CEC2005) 2, 966-973, September 2005.

Paterlini S and Krink T., (2006). Differential Evolution and PSO in Partitional Clustering. Computational Statistics and Data Analysis, 50(2006):1220–1247, 2005.

Pham, D.T., Ghanbarzadeh, A., Koç, E., Otri , S., Rahim , S. and Zaidi, M., (2006). The Bees Algorithm – A novel tool for complex optimisation problems. In: Proc. of the 2nd Virtual International Conference on Intelligent Production Machines and Systems (I*PROMS-06), Cardiff, UK, 2006, 454-459.

Rottensteiner, F., Briese, Ch., (2002). "A new method for building extraction in urban areas from high-resolution LIDAR data", International Archives of Photogrammetry and Remote Sensing, Vol. 34, Part 3A, Graz, Austria

Samadzadegan, F., (2004) Object extraction and recognition from LIDAR data based on fuzzy reasoning and information fusion techniques, *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, Vol. XXXV-B3, Istanbul, Turkey

Tao, C., Hu, Y., (2001). A review of post-processing algorithms for airborne LiDAR data. CD-ROM Proceedings of ASPRS Annual Conference, April 23–27, St. Louis, USA.

Tóvári, D., Vögtle, T., (2004). Classification Methods for 3D Objects in Laserscanning Data. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, Vol. XXXV-B3, Istanbul, Turkey

Vosselman, G., B. Gorte, G. Sithole and Rabbani, T., (2004). Recognising Structure in Laser Scanner Point Clouds, International Conference NATSCAN "Laser-Scanners for Fores and Landscpae Assessment – Instruments, Processing Methods and Applications, ISPRS working group VIII/2, Freiburg im Breisgau, Germany.

Wilson H. G., B. Boots, A.A. Millward, (2004). A comparison of hierarchical and partitional clustering techniques for multispectral image classification, in Proceedings of the International Geoscience and Remote Sensing Symposium, pp. 1624-1626, Toronto, Canada, 2002.

Wu B and Shi Z., (2001). "A Clustering Algorithm based on Swarm Intelligence"; In Proceedings of the International Conference on Info-tech and Info-net, pages 58–66, 2001.

Zhong S. and J. Ghosh, (2003). A comparative study of generative models for document clustering; in SIAM Int. Conf. Data Mining Workshop on Clustering High Dimensional Data and Its Applications, San Francisco, CA, May 2003.

BUILDING FOOTPRINT DATABASE IMPROVEMENT FOR 3D RECONSTRUCTION: A DIRECTION AWARE SPLIT AND MERGE APPROACH

Bruno Vallet and Marc Pierrot-Deseilligny and Didier Boldo

IGN - Laboratoire MATIS 2/4 avenue Pasteur - 94165 Saint-Mand Cedex, France
bruno.vallet@ign.fr - http://recherche.ign.fr/labos/matis

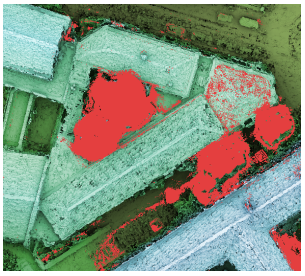
Commission III/3

KEY WORDS: Photogrammetry, 3D reconstruction, building footprint, split and merge, segmentation

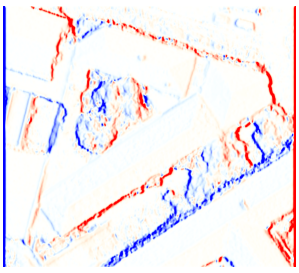
ABSTRACT:

In the context of 3D reconstruction of wide urban areas, the use of building footprints has shown to be of great help to achieve both robustness and precision. These footprints however often present inconsistencies with the data (more than one building in the footprint, inner courts, superstructures...) This paper presents a fast and efficient algorithm to enhance the building footprint database in order to make subsequent 3D reconstructions easier, more accurate and more robust. It is based on a segmentation energy that is minimized by a split and merge approach. The algorithm is demonstrated on a wide urban area of one square kilometer.

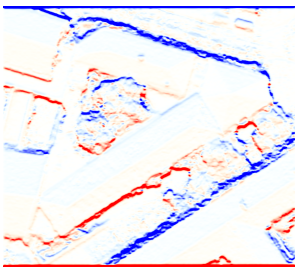
(a) Orthophotography and footprint



(b) Shaded DEM and vegetation mask



(c) Horizontal gradient



(d) Vertical gradient

Figure 1: Input to our algorithm

1 INTRODUCTION

The production of 3D models of urban areas has received a lot of attention from the scientific community in the last decade because of the broad range of its applications and the increase in both quality and quantity of data. In this setup, it becomes more and more crucial to design flexible tools to help human operators achieving efficient and accurate reconstruction of wide urban areas.

1.1 Problem statement

The problem of urban reconstruction consists in finding a 3D model (in general a polygonal surface) that is as coherent as possible with the input data. In our case where the footprints of the buildings are given, we can use the efficient and robust approach proposed in (Durupt and Taillandier, 2006). However, this approach relies heavily on the quality of the building footprint database, and might fail if the building to be reconstructed contains altimetric discontinuities that are not present in its footprint. This often happens in practice, and especially when:

- Two (or more) adjacent buildings with different roof heights share the same footprint.
- The real footprint of a building is only a portion of the footprint in the database (gardens, inner courts,...)
- The building has some superstructures which sizes and heights are not negligeable with respect to the expected precision of the reconstruction. This problem becomes increasingly difficult as reconstructions gain in precision, and has already been tackled in the context of photogrammetry (Bredif et al., 2007) (Dornaika and Bredif, 2008).

More difficult cases are often a combination of the three cited above, and require a manual intervention to enable a further reconstruction. In general, this intervention consists in subdividing the footprint by cutting through all (or most of) the altimetric discontinuities. In a production framework, where large areas need to be extensively reconstructed, it appears that this building footprint database enhancement step is one of the most time consuming. Hence, the problem that we tackle in this paper is that of automatizing this enhancement as a required preprocessing step to 3D reconstruction. More precisely, our problem is to segment a polygonal footprint into a set of non-overlapping polygonal sub-footprints that cover it entirely, such that the interface between the sub-footprints corresponds to altimetric discontinuities. This is a problem of segmentation of vector data (building footprints database) guided by raster data (photos, DEM,...)

1.2 Available data

The data available in our study mainly consisted of:

- A set of 10 centimeter resolution aerial images with a high recovery ratio around 60% (intraband + interband) in order to ensure that each ground point is seen in at least 4 images, covering an area of one square kilometer. The images are in RGBI (the infrared channel is used to obtain the vegetation mask).
- A vectorized cadastral map giving building footprints for the same area. It consists in a set of polygonal footprints given by their ordered list of points in ground coordinates (Figure 1(a), green).

From this initial data, existing algorithms can be run to extract:

- A Digital Elevation Model (DEM) over the whole area (Figure 1(b)). It was obtained by dense correlation following (Roy and Cox, 1998) and the implementation described in (Pierrot-Deseilligny and Paparoditis, 2006).
- The gradient of the DEM (Figures 1(c) and 1(d)) computed using a standard Canny-Deriche filter (Deriche, 1987).
- An orthophotography of the area (Figure 1(a)).
- A vegetation mask (Figure 1(b), red) obtained by the method exposed in (Iovan et al., 2007).

The initial data and extracted data form the *input* to our algorithm.

1.3 Previous works

The idea of using a 2D building footprint to enhance 3D building reconstruction first appeared in (Pasko and Gruber, 1996), and was developed in (Roux and Maitre, 1997), (Brenner, 2000) and (Jibrini et al., 2000). This idea is also at the core of the reconstruction method (Durupt and Taillandier, 2006) for which we designed our building footprint enhancement algorithm, and to the more general framework (Taillandier, 2005) from which it derives. In the context of laser data, it is also central to the works of Vosselman *et. al.* (Vosselman and Dijkman, 2001) (Vosselman and Suveg, 2001) (Suveg and Vosselman, 2001).

To the best of our knowledge, segmentation of building footprints has never been decoupled from the reconstruction itself as done in this paper, but used to find directly planar regions.

1.4 Proposed approach

In this paper we call $\mathcal{P}$ the polygonal footprint to segment, $\mathcal{P}_i$ the polygonal sub-footprint resulting from the segmentation and $I_i^j = \mathcal{P}_i \cap \mathcal{P}_j$ the *interface* between two sub-footprints (it is an edge or set of edges in some cases). The result of our algorithm is a *segmentation* of $\mathcal{P}$ that is given indifferently by the set of sub-footprints $\mathcal{P}_i$ or by the *interface* $\mathcal{I} = \cup_{i<j} I_i^j$ between the $\mathcal{P}_i$ (it is a set of edges).

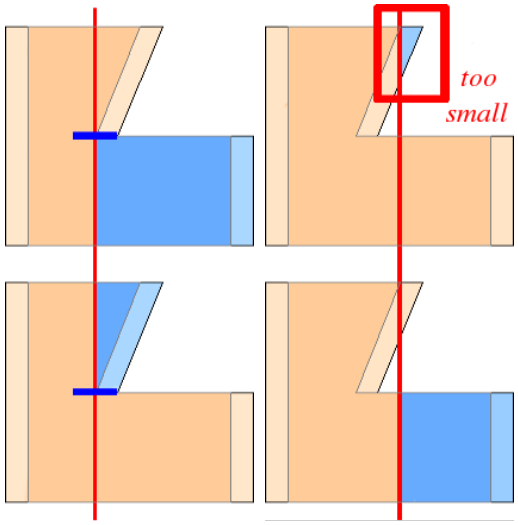
The approach that we propose consists in defining an energy that is negative (resp. positive) on edges that are likely (resp. unlikely) to be altimetric discontinuities, and to find the segmentation that minimizes the sum of this energy over the edges of $\mathcal{I}$. We start by choosing a gradient threshold T_∇ such that we consider that a point where the gradient value is above (resp. below) T_∇ is likely (resp. unlikely) to be on an altimetric discontinuity. The energy on an edge e can then be defined as:

$$E(e) = \int_{P \in e} T_\nabla - |\nabla z(P).\overrightarrow{n}(e)| dP \qquad (1)$$

where z is the height at point P given by the DEM and $\overrightarrow{n}(e)$ is a unit vector normal to e. As required, $E(e)$ is negative when the mean absolute gradient across e is greater than T_∇.

To simplify this problem, and gain in robustness and quality, we will restrict the directions of the interface edges to follow directions present in the original footprint, which is not a strong conditional assumption. This proved to be true on most examples that we have tested. In order to solve this problem, we propose a split and merge approach based on principal directions detected on the initial footprint $\mathcal{P}$:

1. Cluster the directions of the footprint's edges in a direction space taking their lengths into account.
2. Recursively split the footprint along lines of minimal energy.



(a) Two hypotheses for a snapped cut (b) Two hypotheses for a double cut

Figure 2: Cutting hypotheses. The eroded footprint is darkened.

3. Merge the resulting sub-footprints in order to minimize $E(\mathcal{I})$.

The first step is a simple clustering in the space of line angles (modulus π), and does not require special care. Simply notice that we should keep the number of direction clusters as small as possible, for instance by eliminating the clusters which edges' length sum is smaller than a given threshold, or a ratio of the "largest" cluster.

In our algorithm, we will often need to compute energies of the form given by (1) thus to access the gradient across edges that can only be in a limited number of directions. Thus for efficiency reasons, we will precompute the gradient for each direction on a grid aligned with the direction and with the same resolution than the DEM. These grids will serve a double purpose as they will also be used to discretize our cutting lines.

2 RECURSIVE SPLIT

2.1 Cutting hypotheses

For each direction, we will discretize the set of possible cut lines C_i as the lines passing through the (center of) rows of pixels in our grids for each direction. This way the integral of the gradient over an edge in this line's direction will simply be computed as a sum over pixels of the same row in the grid.

As our input footprint might not be convex, a cut might generate more than 2 sub-footprints. In this case, the same cut line C_i generates several cutting hypotheses, one for each edge of $P \cap C_i$ (see Figure 2(b)). Similarly, we snap our cuts by prolongating the initial footprint's edges, and generating a new cut hypothesis for each part of the cut (see Figure 2(a)). This way, each cutting hypothesis consists of the two footprints generated by the split, and their interface I which is a single edge.

This process however can introduce extremely poorly shaped footprints and small footprints that are not desired in the final solution. To prevent the occurrence of such bad geometries, we build an erosion $\mathcal{P}_e$ of the footprint $\mathcal{P}$ by a centered segment of length d orthogonal to the current direction (see Figure 3). This erosion is then used to discard the cutting hypotheses for which:

$$|I \cap \mathcal{P}_e| < |I|/2 \qquad (2)$$

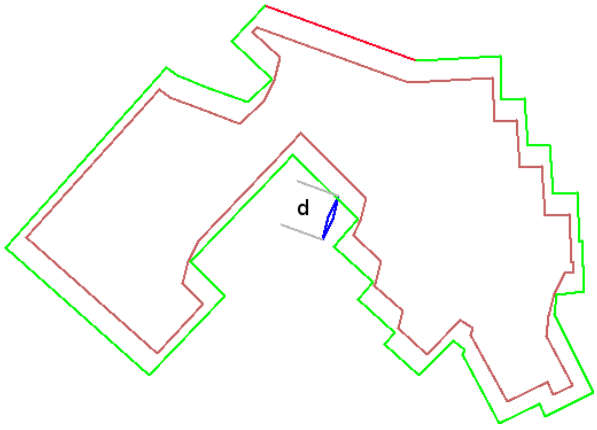


Figure 3: Erosion of the input footprint (green) by a flat rhombus (blue) of height d orthogonal to a main direction (red).

which means that the splitted footprints have a width of at least d on at least half of their length. Hence, the parameter d is used to indicate minimum expected size of a footprint. For instance the hypothesis in Figure 2(b) (top) is discarded because the top right triangle satisfies this criterion (it has $|I \cap \mathcal{P}_e| = 0$). This geometric criterion proved to be the most robust in our experiments, and it was implemented using the CGAL Minkowsky sums. Note that we replaced the segment by a flat rhombus to avoid degeneracies.

2.2 Cut score

For each cut hypostesis, we can compute a cutting score as the energy $E(\mathcal{I})$ restricted to the cut. To enhance this estimation we take into account the following facts:

- An existing edge corresponds to an altimetric discontinuity. Hence the gradient in its vicinity should not be taken into account for the score of a new cut. Thus the meaningful zone is defined by the erosion of the footprint by a centered segment. Ideally, the length of this segment should equal the size of the kernel used to compute the gradient. In practice, it should be even greater as the edges of the footprint are not exactly located on discontinuities. We chose the same length d as before, such that we only need to compute one erosion per footprint and per direction. We chose to compute the erosion with CGAL's exact arithmetics as we encountered failure cases using inexact computations. This is quite time consuming, such that the choice of taking the same parameter is really saving us time.

- Vegetation hides the geometry of the building so the DEM will be considered not pertinent within the vegetation mask.

- The DEM is more inaccurate in shadowed areas.

These three facts are integrated in the computation of $E(\mathcal{I})$ by weighting the gradients by a *confidence* term that is 0 outside the eroded footprint and in vegetation areas, and elsewhere proportional to luminosity.

2.3 Recursion

For the input footprint $\mathcal{P}$, we can build the cutting hypotheses (Section 2.1) and their scores (Section 2.2). We select the cutting hypothesis with the lowest score and apply it to the footprint $\mathcal{P}$, which splits it into two sub-footprints $\mathcal{P}_1$ and $\mathcal{P}_2$. We apply this process again to $\mathcal{P}_1$ and $\mathcal{P}_2$, and so on recursively.

To ensure that our cuts minimize E, we stop the recursion when the lowest score becomes positive. In that case the footprint is *final* and will not be splitted. Our shape criterion (2), guarantees that the width of the resulting sub-footprints is greater than d in each direction.

2.4 Results

As figure 4 shows, the segmentation resulting from the recursive split runs through most of the altimetric discontinuities. However, the segmentation presents many undesired cuts as our cuts are straight so they run through the whole footprint when they may correspond to much more local altimetric discontinuities. To achieve a better segmentation, and further minimize our energy, we need to remove these superfluous cuts by merging sub-footprints whenever this improves the energy $E(B)$.

3 MERGE

3.1 Geometric polygon merging

Merging the sub-footprints resulting from the splitting process can be tricky as numerical precision forces us to use thresholds to determine whether two edges from different polygons touch or not. To make the merge process independent from numerical precision and thresholds, we label all edges produced during the splitting process by (a pointer to) the cut line that produced it. This way, the merging algorithm is both robust and simple:

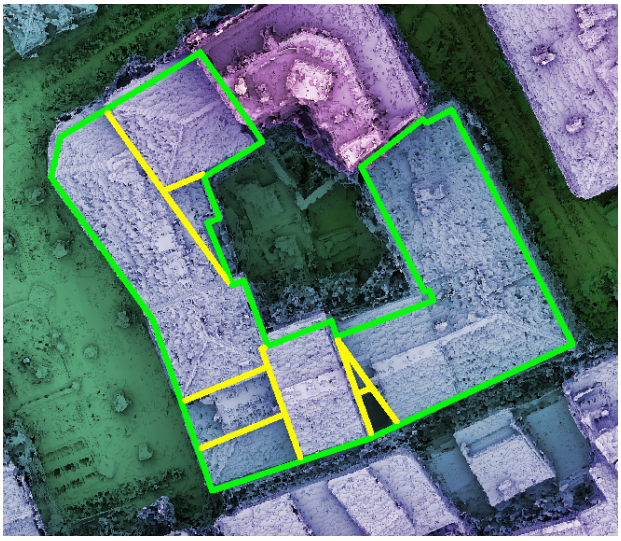
1. For each pair of edges $e_k^i \in P_i$ and $e_l^j \in P_j$ belonging to the same cut line:
   - Compute the intersection edge $e_{k,l} = e_k^i \cap e_l^j$
   - If $e_{k,l} \neq \emptyset$, add $e_{k,l}$ to $I_{i,j}$.
2. Build the connected components of $I_{i,j}$. If there are more than one, this means that the merged footprint has holes. We need to prevent these holes to appear as they are harder to handle in the reconstruction process. To do so, we keep only one connected component in $I_{i,j}$ (the longest or the one with lowest score).
3. Build the merged footprint $P_{i,j}$:
   - For each interface edge $e_{k,l} \in I_{i,j}$ tag e_k^i and e_l^j as interface edges.
   - Build the connected components C_i and C_j of edges of P_i and P_j not tagged as interface.
   - Connect the endpoints of C_i and C_j (this is unambiguous if P_i and P_j where properly oriented).

3.2 Merging algorithm

The merging process goes as follows:

1. Compute all possible merges, their interfaces $I_{i,j}$ and scores $S_{i,j} = E(I_{i,j})$.
2. Build a priority queue of all merges, where the priority is the score $S_{i,j}$. Remember that a high score means it is likely that the interface is not an altimetric discontinuity so it should be removed from the final cut.
3. While the merge with highest priority is positive:
   - Apply the merge with highest priority $S_{i,j}$ between footprints P_i and P_j by replacing P_i and P_j by their union $P_{i,j} = P_i \cup P_j$.
   - Remove all merges involving P_i and P_j from the priority queue.
   - Compute all possible merges involving $P_{i,j}$, their interfaces, their scores, and add them in the priority queue.

(a)

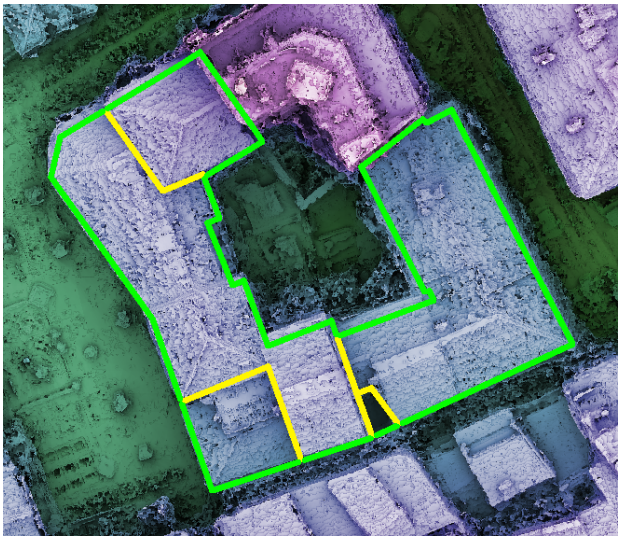


(b)

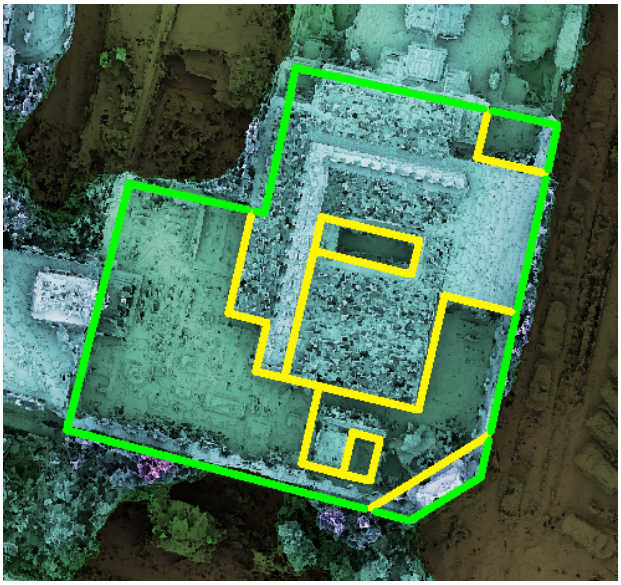
(c)

Figure 4: Results of the splitting process

(a)



(b)



(c)

Figure 5: Results of the merging process

3.3 Results

The merging process ensures that the result is a valid segmentation of the input footprint into a set of sub-footprints. As seen in Figure 5, the algorithm is general enough to allow for a broad range of possible sub-footprints, while being constrained enough (in particular by the allowed directions and minimum size d) to avoid overly complex shapes. The advantage is that such simple shapes are proper for reconstruction. The inconvenient is that if discontinuities do not follow the detected directions, they will not be detected and lead to inconsistencies. Finally, note that as we prevent holes from appearing, inner courts stay connected to the outer boundary (there are two examples of that behavior in Figure 5(c))

4 DISCUSSION

Our method allows for a much more accurate 3D reconstruction on footprints with inner altimetric discontinuities as shown in Figures 6 and 7. However, it sometimes misses some global cuts that are obvious to the eye but do not correspond to altimetric discontinuities. For instance Figure 6 show that a single (and small) handmade cut relying more on a global perception of the footprint shape than on an altimetric discontinuity allows for a great improvement of the result.

This method is proposed as a tool to support the reconstruction of wide urban areas. The splitting and merging results shown here are all obtained based on the same parameters. The tuning parameters are mainly the erosion width d that controls the minimum footprint size and gradient threshold T_∇ that serves to specify the limit between what is a discontinuity and what is not. They are intuitive and simple to tune. In practice, we used the same standard parameters ($d = 1.5m$, $T_\nabla = 3.5$) to process an entire 1km by 1km working area.

Step	(a)	(b)	(c)
Load inputs	0.27	0.4	0.27
Precompute	0.24	0.44	0.12
Erosions	0.2	0.19	0.36
Scores	0.15	0.2	0.1
Splits	0.23	0.17	0.15
Merge	0.01	0.01	0.04
Total	1.1	1.41	1.04

Table 1: Timings (in seconds on a 2.8GHz Pentium 4 processor) of the different steps of the algorithm. The three columns correspond to the examples shown on figures 4 and 5.

In terms of computation time, the algorithm is extremely fast (see table 1). This makes it possible to process very wide working zones rapidly, or to tune the parameters interactively.

The algorithm is heavily dependant on the quality of the input DEM, and only very weakly on the orthophotography and vegetation mask (the latter only serves when the footprint contains vegetation that has an important impact on the DEM, which is quite rare). The most important problems that we encountered are:

- The DEM has a poor quality on shadows as it requires a good contrast. As roughly half of the altimetric discontinuities generate a shadow at their bottom, half of the altimetric discontinuities are not accurately represented in the DEM. We simply added a confidence parameter to handle this issue, but we believe some more adequate solutions can be found.

- If the footprint contains an important altimetric discontinuity that is not aligned with one of the clustered direction, it will perturb the splitting as it will add an important factor to the energy of all cuts not exactly orthogonal to it. To limit this effect we penalized wrong gradient directions by weighting the gradient by a factor $\max\left(0, \cos(2(\vec{n}, \vec{\nabla z}))\right)$ that smoothly decreases from 1 (perfect direction) to 0 for angles greater that $\pi/4$.

- Superstructures cause altimetric discontinuities that are often close to or higher than discontinuities between different buildings. Thus they may generate cuts even with a fine tuning of T_∇. A possible remedy would be to implement a superstructure detection such as (Bredif et al., 2007) prior to cutting.

The energy that we use matches closely the Mumford and Shah segmentation formulation (Mumford and Shah, 1989) except that it has no data attachment term. This drawback is inherent to the problem that we pose, and its consequence will be that we lack of a global quality measure. This will sometimes lead to a lack of global coherence, such as missing a small cut that would enhance greatly the reconstruction (see Figure 6). A workaround would be to interact with the reconstruction method, and for instance only split footprints on which the reconstruction is bad (far from the DEM). As this estimation needs to be done many times, this would require the reconstruction to be very fast, which is not the case for the one that we were working with (at least for complex footprints).

The fact that this energy is not necessarily positive makes it impossible to minimize with graph cuts based segmentation where the non-negativity of weights is a fundamental requirement (Kolmogorov and Zabih, 2004). However, this energy is very natural for segmenting with an unknown a priori number of regions, as minimizing this energy will naturally lead to an optimal number of region, without the need to specify a source/sink pair. For instance, not cutting is a solution like any other, and it has its own energy that can be optimal in the case that no segmentation is required (which is the case on many footprints that are adequate for reconstruction without enhancement). In contrast, graph cut energy is always lower for not cutting than for cutting, and the result is in fact the optimum over bipartition. The drawback is that we cannot use the very efficient graph cut algorithm and need a heuristic approach with no guarantee on optimality.

5 CONCLUSIONS AND FUTURE WORK

We have presented an algorithm to split cadastral maps into smaller regions proper for subsequent 3D reconstruction. The algorithm has only be tested for one reconstruction method but the authors believe it might be a useful preprocessing step to any 3D reconstruction method based on the cadastral map or any other vectorial footprint of the building to reconstruct. The algorithm is simple and fast, as it has been designed with the purpose of helping reconstruction of large urban areas.

In the future, we plan on running this algorithm in a production framework to have a better feedback on its large scale usability. We will also look into correcting the DEM in shadowed area, or maybe detection of altimetric discontinuities directly based on correlation in the aerial images. Finally, we will look into less heuristic means of minimizing our energy, especially in the merging phase.

ACKNOWLEDGEMENTS

The work reported in this paper has been performed as part of Cap Digital Business Cluster TerraNumerica project.

In addition, the authors wish to thank Grgoire Maillet for the important feedback on the usability of the algorithm for productive purposes, Mathieu Brédif for his sound scientific advice and Mélanie Durupt for her help with handling the data.

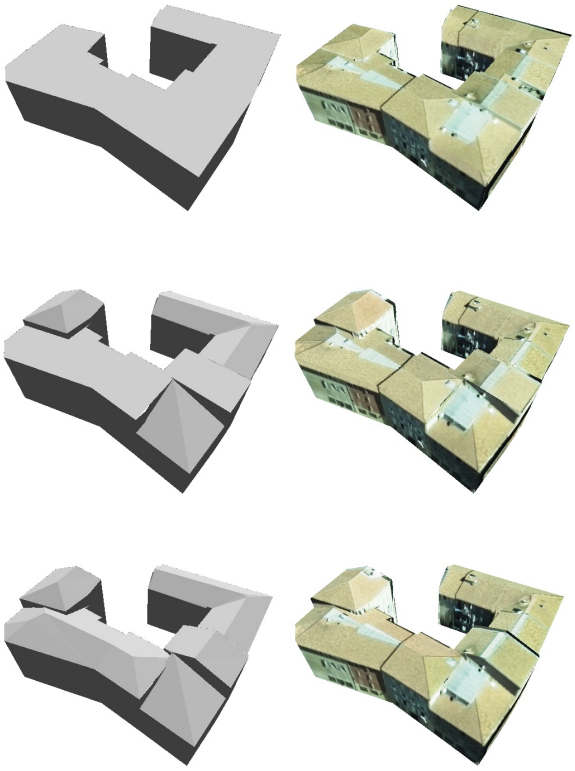


Figure 6: Reconstruction results on the example of Figures 4(b) and 5(b). From top to bottom: reconstruction without enhancement, with enhancement, with enhancement and a single manual cut. This manual cut improves greatly the result but cannot be detected based on our method as the altimetric discontinuity is too low.

REFERENCES

Bredif, M., Boldo, D., Pierrot-Deseilligny, M. and Maitre, H., 2007. 3d building reconstruction with parametric roof superstructures. In: Proc. of the IEEE International Conference on Image Processing.

Brenner, C., 2000. Towards fully automatic generation of city models. In: In: IAPRS, pp. 85–92.

Deriche, R., 1987. Using canny's criteria to derive a recursively implemented optimal edge detector. International Journal of Computer Vision 1(2), pp. 167–187.

Dornaika, F. and Bredif, M., 2008. An efficient approach to building superstructure reconstruction using digital elevation maps. In: IAPRS, Volume 37 (Part 3A).

Durupt, M. and Taillandier, F., 2006. Automatic building reconstruction from a digital elevation model and cadastral data: an operational approach. In: Proc. of the ISPRS Commission III Symposium on Photogrammetric and Computer Vision, ISPRS, Bonn, Germany.

Iovan, C., Boldo, D. and Cord, M., 2007. Automatic extraction of urban vegetation structures from high resolution imagery and digital elevation model. In: URBAN, GRSS/ISPRS Joint Workshop on Data Fusion and Remote Sensing over Urban Areas.

Jibrini, H., Pierrot-Deseilligny, M., Paparoditis, N. and Maitre, H., 2000. Automatic building reconstruction from very high resolution aerial stereopairs using cadastral ground plans. In: Proc. of the XIXth ISPRS Congress, The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, ISPRS, Amsterdam, The Netherlands.

Kolmogorov, V. and Zabih, R., 2004. What energy functions can be minimized via graph cuts? IEEE Transactions on Pattern Analysis and Machine Intelligence.

Mumford, D. and Shah, J., 1989. Optimal approximations by piecewise smooth functions and associated variational problems. Communications on Pure and Applied Mathematics 17(4), pp. 577–685.

Pasko, M. and Gruber, M., 1996. Fusion of 2d gis data and aerial images for 3d building reconstruction. In: Int. Archives of Photogrammetry and Remote Sensing, Vol. XXXI, Part B 3, Vienna.

Pierrot-Deseilligny, M. and Paparoditis, N., 2006. A multiresolution and optimization-based image matching approach: An application to surface reconstruction from spot5-hrs stereo imagery. In: Proc. of the ISPRS Conference Topographic Mapping From Space (With Special Emphasis on Small Satellites), ISPRS, Ankara, Turkey.

Roux, M. and Maitre, H., 1997. Three-dimensional description of dense urban areas using maps and aerial images. In: Extraction of Man-Made Objects from Aerial and Space Images, II, pp. 311–322.

Roy, S. and Cox, I., 1998. A maximum-flow formulation of the n-camera stereo correspondence problem. In: Proc. of the IEEE International Conference on Computer Vision, Bombay, India, pp. 492–499.

Suveg, I. and Vosselman, G., 2001. 3d building reconstruction by map based generation and evaluation of hypotheses. In: Proceedings of the British Machine Vision Conference, p. 643652.

Taillandier, F., 2005. Automatic building reconstruction from cadastral maps and aerial images. In: U. Stilla, F. Rottensteiner and S. Hinz (eds), Proc. of the ISPRS Workshop CMRT 2005: Object Extraction for 3D City Models, Road Databases and Traffic Monitoring - Concepts, Algorithms and Evaluation, Vienna, Austria, pp. 105–110.

Vosselman, G. and Dijkman, S., 2001. 3d building reconstruction from point cloud and ground plans. In: Proc. of the ISPRS Workshop on land surface mapping and characterization using laser altimetry, International Archives of Photogrammetry and Remote Sensing, Vol. XXXIV, Annapolis, U.S., pp. 37–43.

Vosselman, G. and Suveg, I., 2001. Map based building reconstruction from laser data and images. In: Automatic Extraction of Man-Made Objects from Aerial and Space Images (III), pp. 231–239.

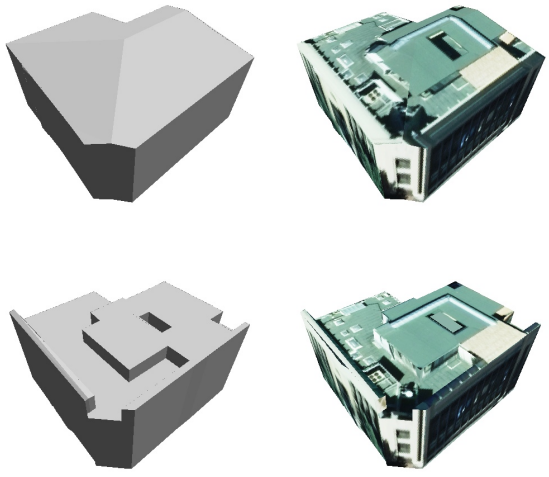


Figure 7: Untextured and textured 3D reconstruction results on the example of Figures 4(c) and 5(c). Top: reconstruction without enhancement, Bottom: with enhancement

A TEST OF AUTOMATIC BUILDING CHANGE DETECTION APPROACHES

Nicolas Champion[a], Franz Rottensteiner[b], Leena Matikainen[c], Xinlian Liang[c], Juha Hyyppä[c] and Brian P. Olsen[d]

[a] IGN, MATIS, Saint-Mandé, France – Nicolas.Champion@ign.fr
[b] Institute of Photogrammetry and GeoInformation,
Leibniz Universität Hannover, Germany – Rottensteiner@ipi.uni-hannover.de
[c] FGI, Dept. of Remote Sensing and Photogrammetry, Masala, Finland –
{Leena.Matikainen, Xinlian.Liang, Juha.Hyyppa}@fgi.fi
[d] National Survey and Cadastre (KMS), Copenhagen, Denmark – bpo@kms.dk

Commission III, WG III/4

KEY WORDS: Change Detection, Building, 2D Vector Databases, Algorithms Comparison, Quality Assessment

ABSTRACT:

The update of databases – in particular 2D building databases – has become a topical issue, especially in the developed countries where such databases have been completed during the last decade. The main issue here concerns the long and costly change detection step, which might be automated by using recently acquired sensor data. The current deficits in automation and the lack of expertise in the domain have driven the EuroSDR to launch a test comparing different change detection approaches, representative of the current state-of-the-art. The main goal of this paper is to present the test bed of this comparison and the results that have been obtained for three different contexts (aerial imagery, satellite imagery, and LIDAR). In addition, we give the overall findings that emerged from our experiences and some promising directions to follow for building an optimal operative system in the future.

1. INTRODUCTION

The production of 2D topographic databases has been completed in many industrialised countries. Presently, most efforts in the National Mapping and Cadastral Agencies (NMCAs) are devoted to the update of such databases. As the update process is generally carried out manually by visual inspection of orthophotos, it is time-consuming and expensive. As a consequence, its automation is of high practical interest for the NMCAs. The update procedure can be split into two steps: *change detection*, in which the outdated database is compared to recently collected sensor data in order to detect changes, and *vectorization*, i.e. the digitization of the correct geometry of the changed objects. Given the state-of-the-art in automatic object detection (Mayer, 2008), only the automation of the change detection step seems to be possible at this time. The key idea is to focus the operator's attention on the areas that may have changed. Work is saved because the operator needs not inspect areas classified as *unchanged* by the automatic procedure.

The current deficits in automation and the lack of expertise within the NMCAs have driven the EuroSDR (European Spatial Data Research - http://www.eurosdr.net) to lauch a project about change detection. It also aims at evaluating the feasibility of semi-automatically detecting changes in a 2D building vector database from optical imagery or LIDAR. Three subtopics are investigated in detail, firstly the impact of methodology; secondly, the impact of the type and spatial resolution of input data; lastly, the impact of the complexity of the scene in terms of interfering objects such as roads. The methodology consists in comparing four different algorithms representative for the current state-of-the-art in the field of change detection. First results, achieved for the cases where only aerial and satellite images are used, were presented in (Champion et al., 2008). The results obtained there showed the limitations of change detection methods, especially in relation to the quality of input data. The main goal of this paper is to present the final results of the project, including a LIDAR dataset, and to give a detailed evaluation of the outcomes delivered by the approaches compared here.

After describing the datasets and the evaluation procedure (Section 2), the methods compared in the test are concisely introduced (Section 3). In Section 4, a thorough evaluation is carried out, including an analysis of the performance of change detection with respect to the update status of the buildings and the building size. The weak and strong points are then identified both for the datasets and the methodologies, and they used to give overall findings and recommendations for building an optimal operative system for change detection in the future.

2. INPUT DATA AND TEST SET-UP

Three test areas are used for the comparison: Marseille (France), Toulouse (France), and Lyngby (Denmark). The area covered by the test sites is 0.9 x 0.4 km^2 in Marseille, 1.1 x 1.1 km^2 in Toulouse, and 2.0 x 2.0 km^2 in Lingby. The test areas differ considerably regarding topography, land use, urban configuration and roofing material. The terrain is hilly in Marseille and Toulouse and relatively flat in Lyngby. Marseille features a densely built-up area consisting of small buildings of variable height, all connected to each other and mostly covered with red tile. Toulouse and Lyngby feature a suburban area, mostly composed of detached buildings and characterised by a large variety of roofing materials such as slate, gravel, or concrete. Colour Infrared (CIR) orthophotos and Digital Surface Model (DSMs) are available for all test areas. In Marseille and Toulouse an image matching algorithm (Pierrot-Deseilligny and Paparoditis, 2006) was used to derive the DSM from input images. In Marseille, these images are multiple aerial images having a forward and side overlap of 60%. The Ground

Sample Distance (GSD) of all input data is 0.2 m. In Toulouse, these images are Pléiades tri-stereoscopic satellite images. The GSD of all input data is 0.5 m. Lastly, the DSM used in Lyngby was derived from first pulse LIDAR data, and the digital orthophoto was generated from a scanned aerial image, both with a GSD of 1 m. For the three test areas, up-to-date vector databases representing the 2D outlines of buildings were available. They served as a reference in the test. In order to achieve an objective evaluation, the outdated databases were simulated by manually adding or removing buildings Thus, 107 changes (out of 1300 buildings in the scene) were simulated in Marseille (89 *new* and 18 *demolished* buildings); 40 (out of 200) in Toulouse (23 *new*, 17 *demolished*) and 50 (out of 500) in Lyngby (29 *new*, 21 *demolished*). The outdated databases were converted to binary building masks having the same GSD as the input data and then distributed to the participants along with input data.

Each group participating in the test was asked to deliver a change map in which each building of the vector database is labelled either as *unchanged*, *demolished* or *new*. Because the methods have been developed in different contexts, their designs noticeably differ, for instance regarding the definitions of the classes considered in the final change map – e.g. four classes for (Champion, 2007) and six classes for (Rottensteiner, 2008) – and the format of the input data – e.g. vector for (Champion, 2007) and raster for (Matikainen et al., 2007). As a work-around, it was decided to use the building label image representing the updated version of the building map (cf. Section 3) for the evaluation of those methods that do not deliver the required change map in the way described above. Only the method by (Champion, 2007) delivered such a change map, which was also directly used in the evaluation.

In order to evaluate the results achieved by the four algorithms, they are compared to the reference database, and the *completeness* and the *correctness* of the results (Heipke et al., 1997) are derived as quality measures:

$$Completeness = \frac{TP}{TP + FN}$$
$$Correctness = \frac{TP}{TP + FP} \quad (1)$$

In Equation 1, *TP*, *FP*, and *FN* are the numbers of True Positives, False Positives, and False Negatives, respectively. They refer to the update status of the vector objects in the automatically-generated change map, compared to their real update status given by the reference. In the case where the final change map is directly used for the evaluation, i.e. with (Champion, 2007), a *TP* is an object of the database reported as *changed* (*demolished* or *new*) that is actually changed in the reference. A *FP* is an object reported as *changed* by the algorithm that has not *changed* in the reference. A *FN* is an object that was reported as *unchanged* by the algorithm, but is *changed* in the reference. In the three other cases, where a building label image representing the updated map is used for the evaluation, the rules for defining an entity as a *TP*, a *FP*, or a *FN* had to be adapted. In these cases, any *unchanged* building in the reference database is considered a *TN* if a predefined percentage (T_h) of its area is covered with buildings in the *new* label image. Otherwise, it is considered a *FP*, because the absence of any correspondence in the *new* label image indicates a change. A *demolished* building in the reference database is considered a *TP* if the percentage of its area covered by any building in the *new* label image is smaller than T_h. Otherwise, it is considered to be a *FN*, because the fact that it corresponds to buildings in the *new* label image indicates that the change has remained undetected. A *new* building in the reference is considered a *TP* if the cover percentage is greater than T_h. Otherwise, it is considered a *FN*. The remaining areas in the *new* label image that do not match any of the previous cases correspond to objects wrongly alerted as *new* by the algorithm and thus constitute *FP*s.

The quality measures are presented in the evaluation on a per-building basis (rather than on a per-pixel basis), as the effectiveness of a change detection approach is limited by the number of changed buildings that is missed or over-detected and not by the area covered by these buildings. As explained in the Section 4, these quality measures are also computed separately for each change class.

3. CHANGE DETECTION APPROACHES

The four methods tested in this study are concisely presented, ordered alphabetically according to the corresponding author.

Champion, 2007: The input of the method is given by a DSM, CIR orthophotos and the outdated vector database. Optionally, the original multiple images can also be used. The outcome of the method is a modified version of the input vector database, in which *demolished* and *unchanged* buildings are labelled and vector objects assumed to be *new* are created. The method starts with the *verification of the database*, where geometric primitives extracted from the DSM (2D contours, i.e. height discontinuities) and, optionally, from multiple images (3D segments), are collected for each object of the existing database and matched with primitives derived from it. A similarity score is then computed for each object and used to achieve a final decision about acceptance (*unchanged*) and rejection (*changed* or *demolished*). The second processing stage, i.e. the *detection of new buildings*, is based on a Digital Terrain Model (DTM) automatically derived from the DSM (Champion and Boldo, 2006), a normalised DSM (nDSM), defined as the difference between the DSM and the DTM, and an above-ground mask, processed from the nDSM by thresholding. Appropriate morphological tools are then used to compare this latter mask to the initial building mask derived from the vector database and a vegetation mask computed from CIR orthophotos and an image corresponding to the Normalised Difference Vegetation Index (NDVI), which results in the extraction of *new* buildings.

Matikainen et al., 2007: The building detection method of the Finnish Geodetic Institute (FGI) was originally developed to use laser scanning data as primary data. In this study, it is directly applied to the input DSM and CIR orthophotos. A raster version of the database (for a part of the study area) is used for training. The method includes three main steps. It starts with segmentation and a two-step classification of input data into *ground* and *above-ground*, based on a point-based analyisis followed by an object-based analysis and using the Terrasolid[1] and Definiens[2] software systems. This is followed by the definition of training segments for buildings and trees and the classification of the *above-ground* segments into *buildings* and *trees*. This classification is based on predefined attributes and a classification tree (Breiman et al., 1984). A large number of

[1] http://www.terrasolid.fi/. Last visited: 30 June 2009.
[2] http://www.definiens.com/. Last visited: 30 June 2009.

attributes can be used, e.g. mean values, standard deviations, texture and shape of the segments. The method automatically selects the most useful ones for classification. In the Marseille area, the criteria selected in the tree included only the NDVI. In the Lyngby area, NDVI and a shape attribute were selected. The third stage consists of a post-processing step that analyses the size and neighborhood of building segments and corrects their class accordingly. Building detection results in a building label image which is used for the comparison in our test.

Olsen and Knudsen, 2005: The input of the method is given by a DSM, CIR orthophotos and a raster version of the outdated database. The method starts with the generation of a DTM, estimated from the DSM through appropriate morphological procedures, a nDSM and an Object Above Terrain (OAT) mask. This is followed by a two-step classification that aims at distinguishing *building* from *no building* objects. This classification is based on criteria that best characterise buildings (especially in terms of size and form) and results in the building label image that is used for the evaluation in this study. The last stage is the actual change detection step, in which the classification outcomes is compared to the initial database in order to extract a preliminary set of potential changes (on a per-pixel basis) that is then post-processed in order to keep only the objects that are assumed to have changed.

Rottensteiner, 2008: This method requires a DSM as the minimum input. Additionally it can use an NDVI image, height differences between the first and the last laser pulse, and the existing database, available either in raster or vector format. The workflow of the method starts with the generation of a coarse DTM by hierarchical morphological filtering, which is used to obtain a nDSM. Along with the other input data, the nDSM is used in a Dempster-Shafer fusion process carried out on a per-pixel basis to distinguish four object classes: *buildings*, *trees*, *grass* land, and *bare soil*. Connected components of *building* pixels are then grouped to constitute initial building regions and a second Dempster-Shafer fusion process is performed on a per-region basis to eliminate remaining trees. Finally, there is the actual change detection step, in which the detected buildings are compared to the existing map, which produces a change map that describes the change status of buildings, both on a per-pixel and a per-building level. Additionally, a label image corresponding to the *new* state of the data base is generated. In spite of the thematic accuracy of the change map produced by this method, it was decided to use this building label image for the evaluation in this test.

4. EVALUATION AND DISCUSSION

In our opinion, the effectiveness of a change detection system is related to its capacity to guide the operator's attention only to objects that have changed so that *unchanged* buildings do not need to be investigated unnecessarily. These considerations result in the evaluation criteria used in this paper to analyze the change detection performance. On the one hand, to support the generation of a map that is really up-to-date, i.e. to be effective qualitatively, the *completeness* of the system for buildings classified as *demolished* and the *correctness* for *unchanged* buildings are required to be high. The *completeness* of *new* buildings also has to be high if the operator is assumed not to look for any *new* building except for those which are suggested by the system. (Note that this also holds true for *modified* buildings, a case not considered in this study because the simulated changes only consisted in *new* and *demolished* buildings). On the other hand, to reduce the amount of manual work required by the operator i.e. to be effective economically, the *correctness* of the changes highlighted by the system and the *completeness* of *unchanged* buildings must be high. However, if a low *completeness* of *unchanged* buildings implies that many buildings are checked uselessly, this is not necessarily critical for the application itself, because the updated database is still correct. Moreover, the economical efficiency that could then appear to be low has to be put into perspective according to the size of the building database to update. For instance, if a change detection system reports 60% of a *national* database as changed, we cannot necessarily conclude about the inefficiency of this system because it still means that 40% of the buildings need not be checked, which amounts to millions of buildings.

4.1 Overall Analysis

Figure 1 presents the evaluation of the results achieved by the methods that processed the Lingby test area (LIDAR context). Table 1 gives the per-building completeness and correctness, obtained for each test area and each approach. The T_h parameter (cf. Section 2.) was set to 0.20 for the Marseille and Lyngby test areas and 0.26 for the Toulouse test area. In Table 1, the values in bold indicate for which methods the best results are achieved. The completeness of detected changes is high for all the methods, especially in the aerial (Marseille) and LIDAR (Lyngby) contexts. By contrast, the correctness observed in our experiments is relatively poor, which indicates that there are many *FP* changes reported by the systems. In this respect, only the results obtained in the Lyngby test area with (Rottensteiner, 2008) seem to achieve a relatively acceptable standard.

Approach	Completeness	Correctness
Marseille (Imagery – Aerial context)		
(Champion, 2007)	94.1%	45.1%
(Matikainen et al., 2007)	**98.8%**	54.3%
(Rottensteiner, 2008)	95.1%	**59.1%**
Toulouse (Imagery – Satellite context)		
(Champion, 2007)	78.9%	**54.5**%
(Rottensteiner, 2008)	**84.2%**	47.1%
Lyngby (LIDAR context)		
(Matikainen et al., 2007)	94.3%	48.8%
(Olsen and Knudsen, 2005)	**95.7%**	53.6%
(Rottensteiner, 2008)	91.4%	**76.1%**

Table 1. Completeness and Correctness achieved by the four algorithms for the three datasets.

To take the analysis further, we also determined the quality measures separately for *unchanged*, *demolished* and *new* buildings. They are presented in Tables 2 (Marseille), 3 (Lyngby) and 4 (Toulouse), respectively. Focusing on the Marseille test area first, it can be seen in Table 2 that all algorithms are effective in detecting the actual changes. Thus, (Matikainen et al., 2007) and (Rottensteiner, 2008) achieve a completeness of 100% for *demolished* buildings. The correctness for *unchanged* buildings is also 100%. The few (11.1%) *demolished* buildings missed by (Champion, 2007) are caused by extracted primitives that are erroneously used in the verification procedure. All three methods also feature a high completeness for *new* buildings. Here, (Matikainen et al., 2007) performs best, with only 2.4% of the *new* buildings missed. The main limitation of this context appears to be the poor correctness rate achieved for *demolished* buildings, which

(a) Matikainen et al. (2007) (b) Rottensteiner (2008) (c) Olsen and Knudsen (2005)

Figure 1. Evaluation of change detection in Lyngby, for (a), (b) and (c) Green: TP; red: FN; orange: FP; blue: TN.

	Unchanged	*Demolished*	*New*
(Champion, 2007)			
Completeness [%]	93.5	88.9	95.2
Correctness [%]	99.8	18.4	63.5
(Matikainen et al., 2007)			
Completeness [%]	94.7	100	97.6
Correctness [%]	100	23.7	75.6
(Rottensteiner, 2008)			
Completeness [%]	94.1	100	94.0
Correctness [%]	100	22.0	96.3

Table 2. Completeness and correctness for the Marseille test area, depending on the update status.

	Unchanged	*Demolished*	*New*
(Matikainen et al., 2007)			
Completeness [%]	81.7	100	91.8
Correctness [%]	100	22.6	100
(Olsen and Knudsen, 2005)			
Completeness [%]	87.8	100	93.9
Correctness [%]	100	30.4	82.1
(Rottensteiner, 2008)			
Completeness [%]	95.9	100	87.8
Correctness [%]	100	56.8	91.8

Table 3. Completeness and correctness for the Lyngby test area, depending on the update status.

	Unchanged	*Demolished*	*New*
(Champion, 2007)			
Completeness [%]	82.8	100	75.0
Correctness [%]	100	42.9	65.2
(Rottensteiner, 2008)			
Completeness [%]	80.2	86.7	82.6
Correctness [%]	97.9	36.1	59.4

Table 4. Completeness and correctness for the Toulouse test area, depending on the update status.

ranges from 18.4% with (Champion, 2007) to 23.7% with (Matikainen et al., 2007). The situation is a bit better for *new* buildings, with a correctness rate larger than 63% for all the methods and even rising to 96.8% with (Rottensteiner, 2008). In spite of such limitations, all the methods presented here are very efficient in classifying *unchanged* buildings, for which the completeness rates are higher than 93%, which indicates that a considerable amount of manual work is saved and also demonstrates the economical efficiency of these approaches in the context of aerial imagery.

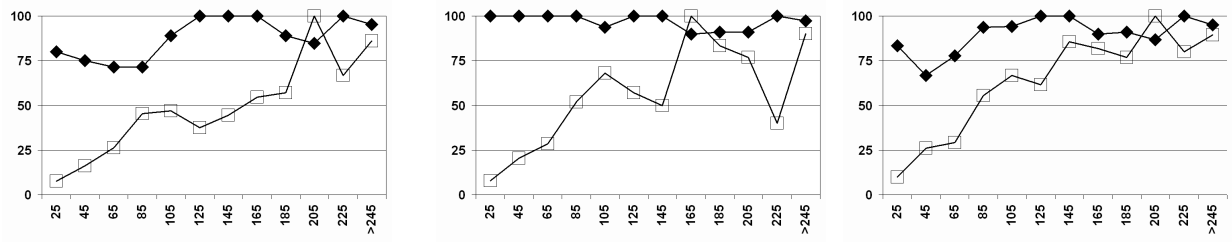
Analyzing Table 3 leads to similar conclusions for the LIDAR context. The correctness rate for the reported *demolished* buildings are again poor and only (Rottensteiner, 2008) achieves less than 50% false positives. However, the methods are very effective in detecting *demolished* buildings and achieve a completeness rate of 100% for this class. Compared to the outcomes obtained in Marseille, the main difference concerns the *new* buildings, which appear to be more difficult to extract. Thus, between 6.1% (Olsen and Knudsen, 2006) and 12.2% (Rottensteiner, 2008) of the *new* buildings are missed. If these percentages of missed *new* buildings can be tolerated, our tests indicate that LIDAR offers a high economical effectiveness and thus may be a viable basis for a future application. If these error rates for *new* buildings are unacceptable, manual post-process is required to find the missed buildings, at the expense of a lower economical efficiency.

The situation is not quite as good with the satellite context (Table 4). The method by (Champion, 2007) is very effective in detecting *demolished* buildings (100%), but this is achieved at the expense of a low correctness rate (42.9%). The same analysis can be carried out with (Rottensteiner, 2008), but this method even misses quite a few *demolished* buildings. It has to be noted that, even though the completeness rates for *unchanged* buildings achieved by both methods are relatively low compared to those obtained in the Marseille and Lyngby test areas, they also indicate that even under challenging circumstances, 80% of *unchanged* buildings need not be investigated by an operator. The main limitation appears to be the detection of *new* buildings. As illustrated for an example in Figures 3e and 3f, 17.4% and 25% of *new* buildings are missed with (Rottensteiner, 2008) and (Champion, 2007) respectively, which is clearly not sufficient to provide a full update of the database and requires a manual intervention in order to find the remaining *new* buildings.

In order to obtain deeper insights into the reasons for failure, in the subsequent sections we will focus our analysis on some factors that affect the change detection performance.

4.2 Impact of the Size of a Change

To analyse the performance of change detection as a function of the change size, we compute the completeness and correctness rates depending on this factor. For that purpose, *new* and *demolished* buildings are placed into bins representing classes



(a) Champion, 2007 (b) Matikainen et al., 2007 (c) Rottensteiner, 2008

Figure 2. Completeness (diamonds) and correctness (squares) of the detection results as a function of the building size [m²].

of 20 m² in width. Note that the buildings from *all* the test areas for which results were submitted are combined in order to have a significant number of changes for each bin. The graphs for (Champion, 2007) and (Rottensteiner, 2008) also contain the results from the Toulouse area. The completeness and correctness rates, computed independently for each bin, are presented in Figure 2 and demonstrate the close relation between the quality of change detection and the change size. This is true for the completeness with (Champion, 2007) and (Rottensteiner, 2008), but it is even more obvious for the correctness in all three graphs. Correctness is particularly poor for buildings smaller than 100 m². Looking at these graphs it becomes obvious that the two major problems observed in Section 4.1, namely the potentially critical rate of missed *new* buildings, which limits the qualitative effectiveness of change detection, and the poor correctness for *demolished* buildings are caused by the same underlying phenomenon i.e. the fact that small objects cannot be detected reliably by an automated procedure. Attentive readers may also notice that a very low correctness occurs with (Matikainen et al., 2007) with buildings covering about 235m². It is caused by large ground areas in the Marseille test area that were mistakenly classified as above-ground objects and then wrongly alerted as *new* buildings.

4.3 Impact of the Quality of the Input Data

Our experiments show that many *FP* cases are related to the quality of the input DSM. The *correlation* DSMs used in the imagery context contain a lot of erroneous height values, especially in shadow areas (where stereo-matching algorithms are known to have problems) that are almost systematically alerted as *new* buildings, as depicted in Figures 3a, 3b, 3c, and 3d. These errors contribute to lower the correctness rate, especially for new buildings, which drops to 63.5% with (Champion, 2007) in Marseille. The high rate of 96.3% obtained here with (Rottensteiner, 2008) may be related to the use of the initial description of the database as a priori information for producing and improving the building label image. In Toulouse, *FP new* buildings were also related to DSM errors, caused by repeating patterns. Another problem concerns the quantisation effects i.e. the fact that the numerical resolution of height values in the *correlation* DSM is restricted to the GSD, which for instance prevents the use of surface roughness as an input parameter for the Dempster-Shafer fusion process in (Rottensteiner, 2008) and ultimately contributes to lower the correctness rate for *demolished* buildings.

Regarding the Lyngby test area, it was a problem that the original data were not available. Single points inside water areas were not eliminated from the data, but used in an interpolation process based on a triangulation of the LIDAR points, producing essentially meaningless data in these water areas that for example caused *FP new* cases with (Olsen and Knudsen, 2005). The other problem was that first pulse (rather than last pulse) data were provided, which caused *FP*s in areas with dense vegetation, e.g. along rivers with (Rottensteiner, 2008). Combined with a relatively low resolution (1 m), these problems contribute to lower the correctness of the systems.

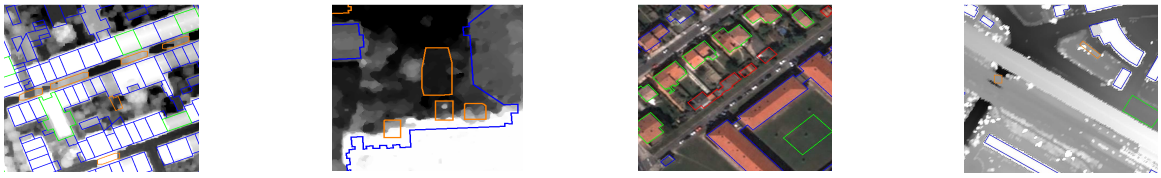
4.4 Impact of Other Topographic Objects in the Scene

In our experiments, some confusion occurs between buildings and other above-ground objects that are present in the scene and wrongly alerted as *new* buildings. Again, this contributes to lower the correctness achieved for *new* buildings. The methods deal with this problem, but currently they only focus on one class of above-ground objects that is to be separated from buildings, namely trees. In general, these trees are identified with indicators based on the NDVI and then eliminated, as shown in Section 3. Even though this strategy appears to be efficient, our experiments show that such confusions are not limited to vegetation but concern other objects that not considered in the approaches presented in this study. For instance, bridges or elevated roads are highlighted as *FP new* buildings in the Lyngby test area by (Rottensteiner, 2008) and (Olsen and Knudsen, 2005), as shown in Figures 3g and 3h. To limit the impact of these problems, two strategies could be considered in the future. The first one consists in developing more sophisticated methods that are capable of simultaneously extracting multiple object classes such as buildings, roads, and vegetation. Such methods would need to incorporate complex scene models that also consider the mutual interactions of the object classes in a scene. They could make use of recent developments in the field of Computer Vision that are related to the modelling context in image classification (Kumar and Hebert, 2006). The second strategy consists in using additional information on other objects, e.g. by incorporating an existing road database in the building change detection procedure.

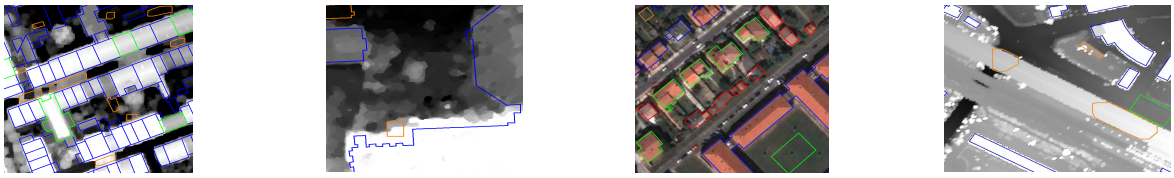
Additional Remark: Beyond the statistical aspects, our experiments show that the errors generated by the change detection approaches are often identical. Thus, the *FP* cases that occur in the Marseille test area because of the DSM inaccuracies (Section 4.3) are both present in the outcomes of (Matikainen et al., 2007) and (Champion, 2007), as illustrated in Figures 3a and 3b respectively. Some of other errors shared at least by two approaches are also illustrated in Figure 3.

5. CONCLUSION

Four building change detection approaches have been tested in three different contexts. If the satellite context appears to be the most challenging for the current state-of-the-art, the aerial context and the LIDAR context appear to be a viable basis for building an operative system in the future. Thus, the high



(a) Matikainen et al., 2007 (c) Rottensteiner, 2008 (e) Rottensteiner, 2008 (g) Rottensteiner, 2008



(b) Champion, 2007 (d) Champion, 2007 (f) Champion, 2007 (h) Olsen and Knudsen, 2005

Figure 3. Evaluation Details (same colour code as Figure 1). *FP new* cases related to DSM errors (shadow areas), in Marseille streets (a)-(b) and Toulouse (c)-(d); (e)-(f) *FN new* cases (small changes); (g)-(h) *FP* new buildings related to bridges.

completeness rates for *demolished* buildings and the high correctness for *unchanged* buildings that could be achieved in these contexts highlight the effectiveness of the presented approaches in verifying the existing objects in the databases. The main limitation in terms of qualitative efficiency concerns the relatively high number of *FN new* buildings – up to 12.1% in the Marseille test area with (Rottensteiner, 2008) – that are mostly related to the object change size. The economical efficiency of the presented approaches seems to be promising, with 80-90% of the existing buildings requiring no further attention by the operator. These buildings are reported to be *unchanged*, which saves a considerable amount of manual work. In terms of the economical efficiency, the main limitation is a high number of *FP demolished* buildings that have to be inspected unnecessarily. Again, this is mainly caused by problems in detecting small changes.

Areas of improvement should concern input data and methodologies. Thus, the resolution of LIDAR data (1 point / m²) used in this test appeared to be critical for the change detection performance: using higher density LIDAR data (e.g. 5-10 points / m²) should improve the situation. As far as methodology is concerned, new primitives should be used in the algorithms, in particular 3D primitives (representing e.g. the 3D roof planes or building outlines) that can now be reliably reconstructed with the 3D acquisition capabilities, offered by recent airborne/spaceborne sensors. Another concern should be the improvement of the scene models used in object detection such that they can deal with different object classes and their mutual interactions. By incorporating different object classes and considering context in the extraction process, several object classes could be detected simultaneously, and the extraction accuracy of all interacting objects could be improved.

In this project, we learned how difficult it is to compare approaches of very different designs. To carry out a fair test, we chose to use the building label images and to limit the type of changes to *demolished* and *new* buildings. In addition, we chose to compare the building label images to the initial vector database, basing on a coverage rate featured by the parameter T_h. Further investigations are necessary to study the actual impact of this parameter on the completeness and correctness rates. However, if we are aware of these drawbacks, we think that this scheme was sufficient to bring out some interesting findings. We also hope that our results – in conjunction with those of e.g. the ARMURS[3] project – will be helpful to create a nucleus of interested people, both in academia and private sector, and to speed up the progress in the vector change detection field.

REFERENCES

Breiman, L., Friedman, J. H., Olshen, R. A., Stone, C. J., 1984. Classification and regression trees. The Wadsworth Statistics / Probability Series, Wadsworth, Inc., Belmont, California.

Champion, N., 2007. 2D building change detection from high resolution aerial images and correlation Digital Surface Models. In: *IAPRSIS* XXXVI-3/W49A, pp. 197–202.

Champion, N., Boldo, D., 2006. A robust algorithm for estimating Digital Terrain Models from Digital Surface Models in dense urban areas. In: *IAPRSIS* XXXVI-3, pp. 111–116.

N. Champion, L. Matikainen, F. Rottensteiner, X. Liang, J. Hyyppä, 2008. A test of 2D building change detection methods: Comparison, evaluation and perspectives. In: *IAPRSIS* XXXVII – B4, pp. 297-304.

Heipke, C., Mayer, H., Wiedemann, C., Jamet, O., 1997. Automated reconstruction of topographic objects from aerial images using vectorized map information. In: *IAPRS*, XXXII, pp. 47–56.

Kumar, S. and Hebert, M.,, 2006. Discriminative random fields. *International Journal of Computer Vision* 68(2), pp. 179–201.

Matikainen, L., Kaartinen, K., Hyyppä, J., 2007. Classification tree based building detection from laser scanner and aerial image data. In: *IAPRSIS* XXXVI, pp. 280–287.

Mayer, H., 2008. Object extraction in photogrammetric computer vision. *ISPRS Journal of Photogrammetry and Remote Sensing* 63(2008), pp. 213-222.

Olsen, B., Knudsen, T., 2005. Automated change detection for validation and update of geodata. In: *Proceedings of 6th Geomatic Week*, Barcelona, Spain.

Pierrot-Deseilligny, M., Paparoditis, N., 2006. An optimization-based surface reconstruction from Spot5- HRS stereo imagery. In: *IAPRSIS* XXXVI-1/W41, pp. 73–77.

Rottensteiner, F., 2008 Automated updating of building data bases from digital surface models and multi-spectral images. In: *IAPRSIS* XXXVII – B3A, pp. 265-270.

[3] http://www.armurs.ulb.ac.be. Last visited: 30 June 2009.

CURVELET APPROACH FOR SAR IMAGE DENOISING, STRUCTURE ENHANCEMENT, AND CHANGE DETECTION

Andreas Schmitt, Birgit Wessel, Achim Roth

German Aerospace Center (DLR)
German Remote Sensing Data Center (DFD), D-82234 Wessling
Andreas.Schmitt@dlr.de, Birgit.Wessel@dlr.de, Achim.Roth@dlr.de

KEY WORDS: SAR, Imagery, Structure, Extraction, Change Detection, Method, Urban

ABSTRACT:

In this paper we present an alternative method for SAR image denoising, structure enhancement, and change detection based on the curvelet transform. Curvelets can be denoted as a two dimensional further development of the well-known wavelets. The original image is decomposed into linear ridge-like structures, that appear in different scales (longer or shorter structures), directions (orientation of the structure) and locations. The influence of these single components on the original image is weighted by the corresponding coefficients. By means of these coefficients one has direct access to the linear structures present in the image. To suppress noise in a given SAR image weak structures indicated by low coefficients can be suppressed by setting the corresponding coefficients to zero. To enhance structures only coefficients in the scale of interest are preserved and all others are set to zero. Two same-sized images assumed even a change detection can be done in the curvelet coefficient domain. The curvelet coefficients of both images are differentiated and manipulated in order to enhance strong and to suppress small scale (pixel-wise) changes. After the inverse curvelet transform the resulting image contains only those structures, that have been chosen via the coefficient manipulation. Our approach is applied to TerraSAR-X High Resolution Spotlight images of the city of Munich. The curvelet transform turns out to be a powerful tool for image enhancement in fine-structured areas, whereas it fails in originally homogeneous areas like grassland. In the change detection context this method is very sensitive towards changes in structures instead of single pixel or large area changes. Therefore, for purely urban structures or construction sites this method provides excellent and robust results. While this approach runs without any interaction of an operator, the interpretation of the detected changes requires still much knowledge about the underlying objects.

1 INTRODUCTION

Nowadays spaceborne SAR data is easily available. Thanks to the high resolution of up to one meter (TerraSAR-X) it is suitable for urban applications, e.g. urban growth modeling as well as for damage mapping in conjunction with (natural) disasters. A main problem for SAR image interpretation apart from the geometrical aspect is the high noise level caused by the combination of deterministic (speckle effect) and random noise. The reduction of noise, e.g. by the multi-looking approach, often goes along with a loss of resolution. While structure preserving filters do not enhance fine-structured areas, smoothening filters even blur the structures apparent in SAR data over urban areas. So resolution and structure preserving filter algorithms are still a topic of research. In this context alternative image representations like wavelets have been applied. While wavelets are used to separate point singularities (Candès and Donoho, 1999), second generation wavelets, e.g. curvelets, are more suitable for the extraction of two dimensional features, as they are able to describe image discontinuities along a smooth line (an edge) with a minimum number of coefficients (Candès and Donoho, 1999). The elementary components are the so-called ridgelets – due to their appearance like a ridge – that can have different scales (equivalent to their length), directions and positions in the image. This enables a selection of two dimensional features to be suppressed (assumed noise) or to be emphasized (structure) by manipulating the corresponding coefficient of each ridgelet. In the following a short overview to related work especially to the development of curvelets is given. Then, the curvelet representation is roughly explained and three applications are presented: image denoising, structure enhancement and change detection over the city center of Munich (imaged by TerraSAR-X in the high resolution spotlight mode and VV polarization). So this paper shows the potential of the curvelet transform for SAR image analysis.

2 RELATED WORK

The curvelet transform used in this approach has originally been developed by (Candès and Donoho, 1999) to describe an object with edges with a minimal number of coefficients in the continuous space. Much research work was done to examine the behaviour of curvelets (Candès and Donoho, 2002a, Candès and Demanet, 2002b, Candès and Guo, 2002), to transfer the definitions from the continuous to the discrete space (Candès and Donoho, 2003a, Candès and Donoho, 2003b) and to accelerate the computing time (Candès et al., 2005) so that digital image processing becomes feasible. Many applications in different scientific fields have been published so far, e.g. in geo- and astrophysics, that are summarized on the curvelet homepage (Demanet, 2007).

Denoising of SAR images to simplify image analysis has also been a research topic during the last years where many approaches have been published. (Ali et al., 2007) proposed a combination of a wavelet based multi-scale representation and some filters to improve the results obtained by the ”standard” filtering techniques like the Lee-filter. A bayesian-based method using ”a trous” filter in the wavelet domain has been proposed by (Moghaddam et al., 2004). Because of the properties of the wavelet transform, originally developed for one dimensional data, these two methods are able to smooth regions and to suppress point-like noise, but they do not take into account the two dimensional nature of images. The advantage of second generation wavelets for despeckling has been examined by (Gleich et al., 2008) for the bandelet and the contourlet transform. The application of curvelets on optical and ultrasound images respectively in the medical context has been published by (Ma et al., 2007). The only publication on the use of curvelets in the remote sensing context by (Sveinsson and Benediktsson, 2007) presents a denoising technique with a

combination of wavelets and curvelets. A total variation based segmentation algorithm divides the image in structured regions, that are subsequently denoised by a curvelet-based method, and homogeneous regions, denoised by a wavelet approach. For large scenes with different land cover types, this method seems to be very promising. As we concentrate on urban applications in this paper, we use a purely curvelet-based approach.

Change detection in SAR images being a very difficult task has often been discussed in literature. An overview to principal SAR change detection methods, their advantages as well as their disadvantages can be found in (Polidori et al., 1995). Some more specialized methods are touched in the following. The approach of (Balz, 2004) uses a high resolution elevation model (e.g. acquired by airborne laserscanning) to simulate a SAR image which is subsequently compared to the real SAR data. The quality of the results is naturally highly dependent on the resolution of the digital elevation model and its co-registration to the SAR image. This nontrivial co-registration constraints this approach to small scale exemplary applications. Another idea starting with the fusion of several SAR images of different incidence angles to a "superresolution" image is presented by (Marcos et al., 2006) and (Romero et al., 2006). Man-made objects, i.e. geometrical particularities that are not captured by the digital terrain model used for the orthorectification of the SAR image, are classified by their diverse appearance in the single orthorectified images due to the different acquisition geometries. So, seasonal changes in natural surroundings can easily be distinguished from changes in built-up areas. One disadvantage is the large number of different SAR images of the same area needed to generate the "superresolution" image. (Wright et al., 2005) exploits the coherence (phase information) of two SAR images, which implies a relatively short repeat-pass time to avoid additional incoherence caused by natural surfaces. (Derrode et al., 2003) and (Bouyahia et al., 2008) adopt a hidden and a sliding hidden Markov chain model respectively to select areas with changes in reflectivity even from images with different incidence angles. Although this method allows to process very large images and does not need additional parameter tuning, except the window size, according to the authors still a lot of research work has to be done to improve the preliminary results.

3 CURVELET REPRESENTATION

The curvelet representation consists of three components according to (Candès and Donoho, 1999):

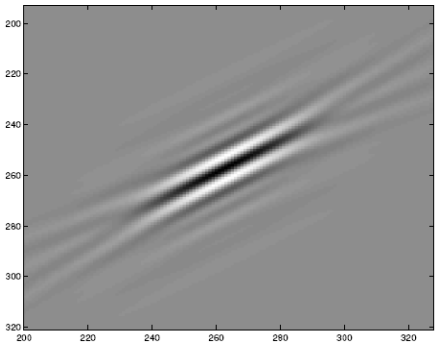
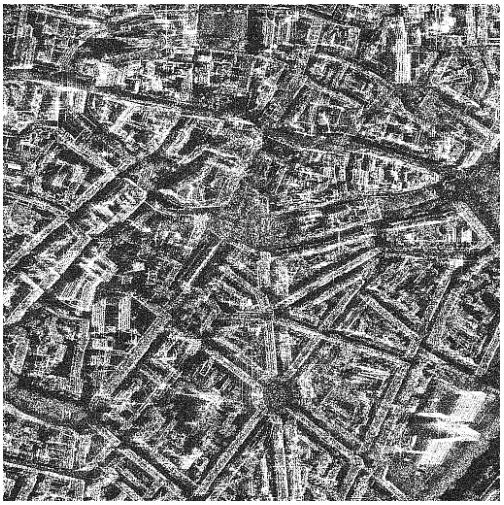


Figure 1: Ridgelet in spatial domain (Candès et al., 2005)

Ridgelets These two dimensional waveforms are the basic elements of the curvelet representation. In the spatial domain, they appear like a ridge or a needle (see Fig. 1); in the curvelet domain their contribution to the original image is measured by a coefficient. The magnitudes of the ridgelets extracted from Fig. 2(a) are depicted in Fig. 2(b) by gray-values. Bright pixels mark high magnitudes. In contrast to wavelets, curvelets are additionally defined by their orientation in the two dimensional space (Ying et al., 2005). Hence, this is a method of image analysis suitable for image features with discontinuities across straight lines.



(a) Spatial domain

(b) Curvelet coefficients

Figure 2: City center of Munich, imaged by TerraSAR-X, High Resolution Spotlight mode, Polarisation VV, Spatially Enhanced Multi Look Ground Range Detected product

Multiscale ridgelets As the decomposition into ridgelets is dependent on the scale, a pyramid of windowed ridgelets is used, renormalized and transported to a wide range of scales and locations. For example, a ridgelet on the finest scale (N4-neighborhood) can only be horizontally or vertically oriented, i.e. two different orientations, while a ridgelet on the next coarser scale has already twice as much, i.e. four different orientations. Consequently, the resolution in orientation increases with coarser ridgelet scales. The number of directions is given by the formula $2^{subband}$. For redundancy reduction a wavelet decomposition is commonly used on the finest scale, where only horizontal and vertical directions are discriminable anyway (Candès et al., 2005). The different scales appear in Fig. 2(b) as single rings, whereas the outer rings show the finer scales. The gaps between the rings are just for visualization.

Bandpass Filtering Before the computation of the ridgelets can be done, the original image has to be separated out into a series of disjoint scales. This is done by a Laplacian pyramid which implies a high redundancy in the order of multiplying the original data volume by the factor 16 (Donoho and Duncan, 2000). The interesting thing for images with edges is, that most of these coefficients can be set to zero without loosing any structures. So, data volume reduction gets possible although the initial increase.

If one compares the original SAR image (Fig. 2(a)) to the coefficients' magnitudes (Fig. 2(b)) it is recognizable that the main axes of the city center (a cross slightly rotated clockwise to the vertical and the horizontal direction respectively) correspond in their direction with accumulations of brighter points, i.e. with higher coefficients, in the illustration of the curvelet representation. Now, the idea is to manipulate these coefficients to accent certain structures by preserving the related coefficients or to suppress certain structures by removing the related coefficients before the inverse curvelet transform is done to get the enhanced image in the spatial domain.

4 IMAGE ENHANCEMENT

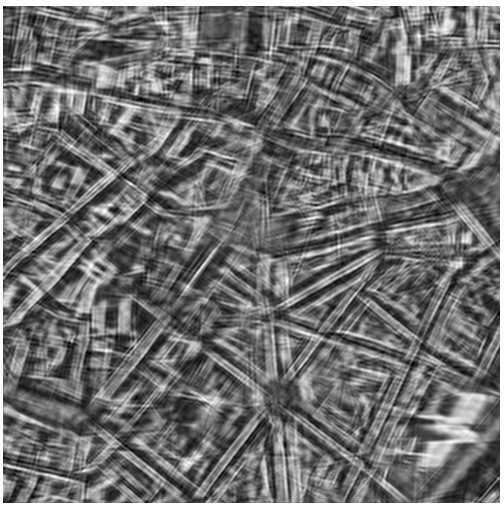
The first application presented here is image enhancement by simple noise suppression and structure extraction respectively.

4.1 Image denoising

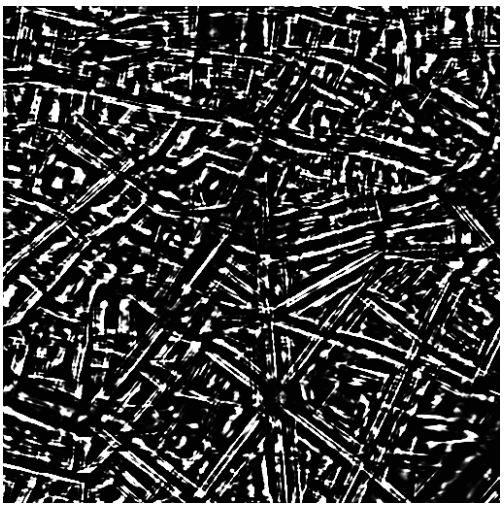
Noise is commonly associated with insignificant curvelet coefficients, therefore a thresholding can set minor coefficients to zero. One problem is that the number of coefficients preserved also corresponds to the complexity of the scene, i.e. if the number of coefficients preserved is defined as constant in advance the complexity of all scenes is seen as equal. By contrast if a magnitude threshold is chosen to exclude minor coefficients, the complexity of the scenes may vary. But in this case the mean magnitude of the coefficients, which is correlated with the contrast in the original image, is misleadingly seen as constant. So, only structures of a certain contrast would be extracted. Fig. 3(a) shows an example where a magnitude threshold of 0.1 was applied, i.e. all lower coefficients were set to zero. It is obvious that the main structures are enhanced, but also many artifacts are produced, that constrain the interpretation. Hence, the determination of a suitable threshold is a difficult task.

4.2 Structure enhancement

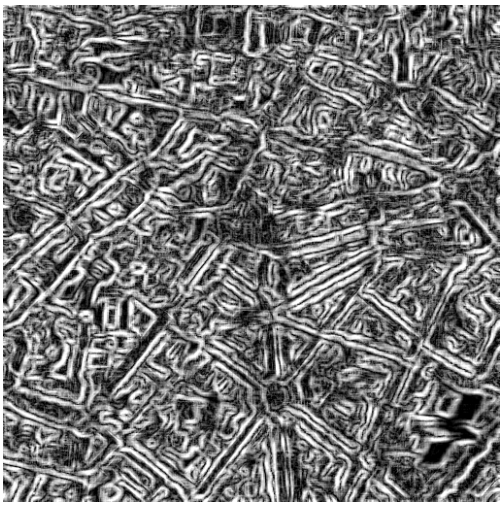
Another possibility is to access the structures via their belonging scale. The finest structures are gray value differences in a N4-neighborhood. As this scale probably only contains noise, all coefficients of this scale are set to zero. The coarsest scale influences the brightness of the image and should be kept unchanged. The scales in-between gather the remaining structures according to their length. So, it is possible to choose only those structures of a certain length to be kept and to suppress all other structures by setting the corresponding coefficients to zero. For example in Fig. 3(b) only the structures of a length from 3 to 300 m are preserved to extract structures that presumably belong to buildings. One can perceive that the main structures of the original image (Fig. 2(a)) are strengthened and all clutter is removed. At first glance the Touzi edge extractor (Fig. 3(c)) and the curvelet approach provide similar results. The lines extracted by the Touzi operator (Touzi et al., 1988) are smoother and closed, but also many lines inside the building blocks are displayed. The important difference between the two approaches is that the curvelet approach only enhances the existing structures while the Touzi extractor traces discontinuities in-between dark and bright structures. Hence, a single linear bright feature on a dark background is strengthened by the curvelet approach, but it is split into two edges by the Touzi extractor.



(a) Reconstructed "denoised" image



(b) Structure reconstruction by curvelets



(c) Touzi edge extractor (r=4)

Figure 3: Denoising and structure extraction of Fig. 2(a)

5 CHANGE DETECTION

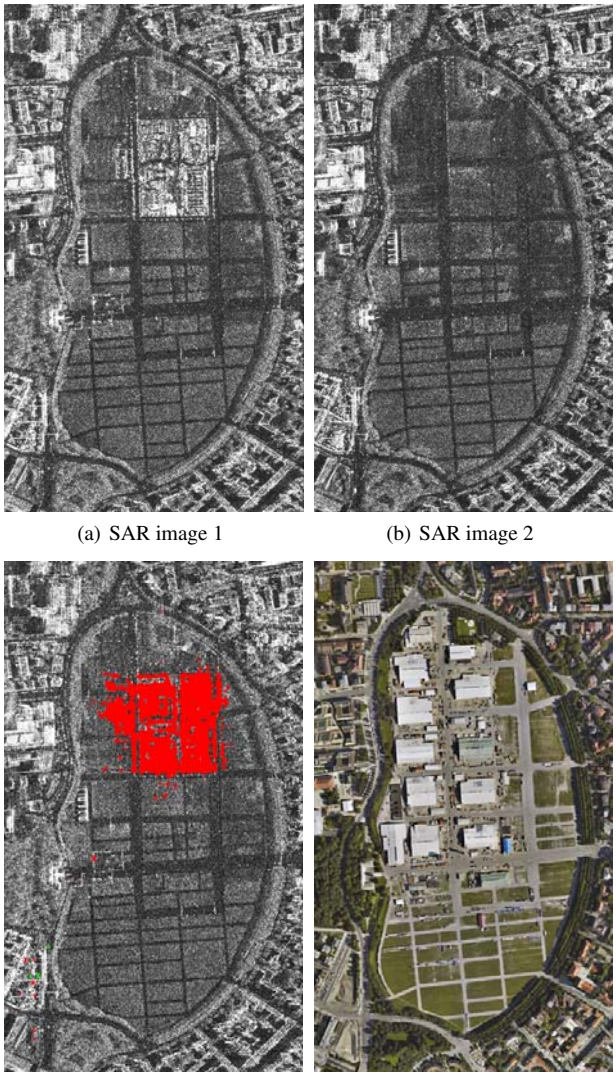
As mentioned before SAR images are highly affected by noise. Although the influence of the deterministic speckle effect should be exactly the same under the same conditions, it is impossible to assure exactly the same conditions over a longer period of time. So, if two SAR images are differentiated pixel by pixel the result is expected to appear very noisy. Alternatively this differentiation can be calculated in the curvelet coefficient domain. If the input images are co-registered and same-sized, the images share also the same combination of curvelet coefficients. Before the difference image is transformed back to the spatial domain, the coefficient differences can be either denoised following Section 4 or weighted quadratically. In the latter case each coefficient is multiplied by its own magnitude to suppress low and to strengthen high coefficients. Additionally the influences of the different scales are equalized by the factor $2^{subband}$ (cf. Section 3). As the resulting image contains positive as well as negative values, the positive values showing regions that brightened up are coded in green and the negative values showing regions that darkened are coded in red. For TerraSAR-X data the geolocation of the detected data product turned out to be sufficient for the change detection, so that no further co-registration was necessary.

A disadvantage of this method might be its high demand on memory. The curvelet representation itself is very redundant increasing the data volume of an image by the factor 16. Although most coefficients are nearly zero or set to zero during the image enhancement process (cf. Section 4), but they have to be processed during the differentiation as well. If more than three images are compared the difference matrix including all relative differences between the input images inflates. But the increasing number of coefficients goes along with an increasing flexibility in approximating linear features in the input image. Tests with other second order wavelets proved that critically sub-sampled approaches do not provide comparable results. To get an impression of the processing time: The example in Section 5.2 including three input images of 2091x1113 pixels are processed with a Matlab implementation and require seven minutes on a Solaris workstation.

In the following two examples over the city of Munich are presented. The first one deals with short time changes in the well-known fairground "Theresienwiese", the second one surveys construction activities near the central station over the period of one year. The processed data sets are acquired by TerraSAR-X in the High Resolution Spotlight mode and delivered as Multi Look Ground Range Detected product.

5.1 Short time changes

The two images of the fairground "Theresienwiese" (Fig. 4(d)) have been acquired in December 2008 and January 2009. Being processed as spatially enhanced product they have a pixel spacing of 0.5 m on ground. Because of the relatively short time lag, the reflectivity of the surrounding is expected to be the same, so all changes should be man-made. Comparing visually the two input images (Fig. 4(a) and 4(b)) one can remark a brighter area in the upper middle of Fig. 4(a) that darkened in the second image (Fig. 4(b)). Especially on the streets inside the fairground many single pixel changes are obvious. For urban applications single pixel changes do only disturb the interpretation as one is more interested in changes happened to structures like streets or buildings. So, these single pixel changes have to be excluded. Spatial averaging would help to find large areas with high changes, but fine linear structures would be smeared over and probably get lost. The curvelet approach is able to preserve the structures while single pixel changes are suppressed. In Fig. 4(c) there



(a) SAR image 1 (b) SAR image 2

(c) Detected changes (d) Optical image ©GoogleEarth

Figure 4: Change detection in the fairground "Theresienwiese" (1: 05.12.2008, 2: 18.01.2009)

is one red region in the upper middle of the image, that accords with the visual interpretation. These changes refer to the "Winter-Tollwood" festival that took place during the first acquisition. The pavilions caused a much higher reflectivity than the bare soil during the second acquisition. Additionally there are some small red and green structures at the bottom left of Fig. 4(c) that were not visible before. Those refer to buses and cars on a parking lot. The slightly darkened region in the middle right of Fig. 4(a) and 4(b) respectively is not marked as change because it does not contain any structure. In summary, the change image shows nearly no disturbances as all small scale changes are excluded. The curvelet approach is very sensitive towards structures (e.g. buses) and very robust towards slight large scale changes caused by environmental influences.

5.2 Long time changes

For damage mapping after natural disasters it is only seldom possible to access up-to-date reference data, as most events cannot be predicted yet. So, seasonal changes in the surrounding of the regions of interest have to be taken into account. The three images of the railway station "Donnersberger Brücke" acquired in March 2008 (Fig. 5(b)), September 2008 (Fig. 5(c)), and March 2009 (Fig. 5(d)) are used to map the construction progress inside the

(a) Optical image ©GoogleEarth

(b) SAR image 1

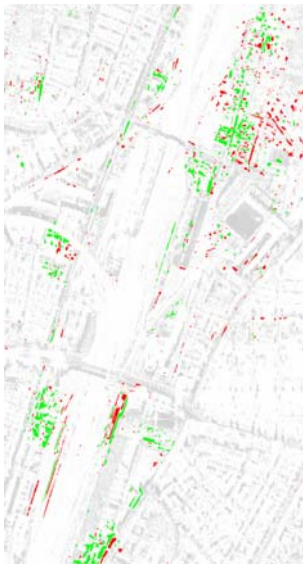
(c) SAR image 2

(d) SAR image 3

Figure 5: Construction site near "Donnersberger Brücke" (1: 30.03.2008, 2: 22.09.2008, 3: 17.03.2009)

(a) Color composite

(b) Detected changes 1 – 3



(c) Detected changes 1 – 2

(d) Detected changes 2 – 3

Figure 6: Change detection (cf. Fig. 5)

construction sites along the railway tracks where new residential and office buildings are planned. As radiometrically enhanced products they share a pixel spacing of 1.25 m on ground. The color composite (Fig. 6(a), 1:R, 2:G, 3:B) shows many colored regions, that help to identify the construction sites. But it is still impossible to interpret these changes. Fig. 6(b) indicates the detected changes by the curvelet approach. Many green structures stand for an increase in reflectivity over the period of one year. A higher reflectivity refers to new objects, e.g. walls or houses while the darkened regions (in red) usually refer to strong scatterers that have disappeared, e.g. scaffoldings. At the bottom left there are sequences of green and red lines which can be interpreted as new buildings. One the one hand a new risen building causes a higher reflectivity (green), on the other hand it also causes new radar shadows (red). Some long green or red lines can be perceived in the middle of the image that refer to trains in the railway depots. Having a look at Fig. 6(c) and 6(d) much more small structures especially at the top right appear. Most of these are marked in red in Fig. 6(c) and in green in Fig. 6(d), so that they compensate each other over the whole year (Fig. 6(b)). These changes are mainly found in the "Hirschgarten" park (see Fig. 5(a) at the top right) comparing the images acquired in spring with those acquired in fall. As these changes are restricted to natural surroundings, they supposedly refer to seasonal changes in the reflectivity by the tree's growth. The blank branches in March cause a much higher reflectivity in the co-polarized channel than the leaves in September. Again the curvelet approach produces a change image with no single pixel disturbances. Changes in the underlying structures are emphasized. Unfortunately it is a difficult task to distinguish man-made changes from seasonal changes in the natural surrounding without a high resolution land cover mask.

6 CONCLUSION

A new approach for SAR image enhancement and change detection based on the curvelet transform has been proposed and applied to TerraSAR-X data of the city center of Munich. As input data any amplitude image can be used, for change detection two equally sized and co-registered images are necessary. Radar

inherent noise is reduced and underlying structures are enhanced depending on their length, their orientation or their intensity.

In the image enhancement context this approach is most suitable for fine-structured areas, e.g. city centers. The main problem lies in the determination of thresholds for suppression and emphasis of structures. The determination of the threshold and the number of coefficients respectively is still experiential and highly dependent on the image content. If the scenes are reconstructed by a fix number of coefficients, the complexity of the scene is restricted. As the image description by the curvelet coefficients is purely based on structures, by omitting coefficients originally smooth areas are often affected by artifacts. At the moment the quadratic weighting of the single curvelet coefficients seems to be the best solution for fully automatic processing chains.

The change detection approach provides excellent results in urban areas. The great advantage over pixel based methods is the sensitivity towards changes in structures and the possibility to predefine the scale and the strength of changes to be mapped. Problems occur in natural surroundings like forested areas, where the status of the foliage has an important seasonal impact on the backscattering behavior. Not to mention the weather conditions, snow cover with different moistures can highly modify the appearance in a SAR image. In consequence of that the interpretation of the detected changes is very challenging. Although the change images contain clear structures without any disturbances, it is nearly impossible to distinguish man-made from natural, e.g. seasonal, changes, without a priori knowledge about the land cover.

As the present results proved that two single polarized SAR images can be used to indicate changes happened to the imaged area, but they do not provide the information needed to interpret these changes, our future research will try to include other data sources into the processing chain. To discriminate natural cover from man-made objects, a coherence layer, that exploits the phase information of the input images could be helpful. Polarimetric layers could facilitate the interpretation by attaching information about the scattering types to the detected changes. Apart from remote sensing data it is quite conceivable to introduce a priori knowledge by overlaying the change layer with land cover classifications from optical data sources as well as with cadastral data sets.

References

Ali, S. M., Javed, M. Y. and Khattak, N. S., 2007. Wavelet-Based Despeckling of Synthetic Aperture Radar Images Using Adaptive and Mean Filters. In: Proceedings of World Academy of Science, Engineering and Technology, Venice (Italy), Vol. 25, pp. 39–43.

Balz, T., 2004. SAR simulation based change detection with high-resolution SAR images in urban environments. In: ISPRS Congress, Istanbul 2004, Proceedings of Commission VII, Vol. 35, pp. 472–477.

Bouyahia, Z., Benyoussef, L. and Derrode, S., 2008. Change detection in synthetic aperture radar images with a sliding hidden Markov chain model. Journal of Applied Remote Sensing (JARS), SPIE.

Candès, E. J. and Demanet, L., 2002b. Curvelets and Fourier integral operators. Compte Rendus de l'Academie des Sciences 336, pp. 395–398.

Candès, E. J. and Donoho, D. L., 1999. Curve and Surface Fitting. Innovations in Applied Mathematics, Vanderbilt University Press, Nashville (TN), Saint-Malo (France), chapter Curvelets – a surprisingly effective nonadaptive representation for objects with edges, pp. 105–120.

Candès, E. J. and Donoho, D. L., 2002a. New Tight Frames of Curvelets and Optimal Representations of Objects with Smooth Singularities. Comm. Pure Appl. Math. 57, pp. 219–266.

Candès, E. J. and Donoho, D. L., 2003a. Continuous Curvelet Transform II: Discretization and Frames. Appl. Comput. Harmon. Anal. 19, pp. 162–197.

Candès, E. J. and Donoho, D. L., 2003b. Continuous Curvelet Transform I: Resolution of the Wavefront Set. Appl. Comput. Harmon. Anal. 19, pp. 198–222.

Candès, E. J. and Guo, F., 2002. New Multiscale Transforms, Minimum Total Variation Synthesis: Applications to Edge-Preserving Image Reconstruction. Signal Processing 82, pp. 1519–1543.

Candès, E. J., Demanet, L., Donoho, D. L. and Ying, L., 2005. Fast Discrete Curvelet Transforms. Multiscale Model. Simul. 5, pp. 861–899.

Demanet, L., 2007. curvelet.org. http://www.curvelet.org. (accessed on 26 March 2009).

Derrode, S., Mercier, G. and Pieczynski, W., 2003. Unsupervised Change Detection in SAR Images Using a Multicomponent HMC model. In: P. Smits and L. Bruzzone (eds), Second International Workshop on the Analysis of Multitemporal Remote Sensing Images, European Commission Joint Research Centre, Ispra (Italy), pp. 16–18.

Donoho, D. L. and Duncan, M. R., 2000. Digital Curvelet Transform: Strategy, Implementation and Experiments. In: Aerosense 2000, Wavelet Applications VII, Orlando (FL), pp. 12–29.

Gleich, D., Kseneman, M. and Datcu, M., 2008. Despeckling of TerraSAR-X data using second generation wavelets. In: ESA EUSC 2008: Image Information Mining, Frascati (Italy).

Ma, L., Ma, J. and Shen, Y., 2007. Pixel Fusion Based Curvelets and Wavelets Denoise Algorithm. Engineering Letters, Online Journal of International Association of Engineers (IAENG) 14(2), pp. 130–134.

Marcos, J.-S., Romero, R., Carrasco, D., Moreno, V., Valero, J. L. and Lafitte, M., 2006. Implementation of new SAR change detection methods: superresolution SAR change detector. In: ESA-EUSC 2006: Image Information Mining for Security and Intelligence, Torrejon air base - Madrid (Spain).

Moghaddam, H. A., Zouj, M. J. V. and Dehghani, M., 2004. Bayesian-based Despeckling in Wavelet Domain Using "a Trous" Algorithm. In: ISPRS Congress, Istanbul 2004, Proceedings of Commission VII, pp. 27–13.

Polidori, L., Caillault, S. and Canaud, J.-L., 1995. Change detection in radar images: methods and operational constraints. In: Geoscience and Remote Sensing Symposium, 1995. IGARSS '95. 'Quantitative Remote Sensing for Science and Applications', Florence (Italy), Vol. 2, pp. 1529–1531.

Romero, R., M. J.-S., Carrasco, D., Moreno, V., Valero, J. L. and Lafitte, M., 2006. SAR Superresolution Change Detection for Security Applications. In: ESA-EUSC: Image Information Mining for Security and Intelligence 2006, EUSC, Torrejon air base - Madrid (Spain).

Sveinsson, J. and Benediktsson, J., 2007. Combined wavelet and curvelet denoising of SAR images using TV segmentation. In: Geoscience and Remote Sensing Symposium, IGARSS 2007, Barcelona (Spain), pp. 503–506.

Touzi, R., Lopes, A. and Bousquet, P., 1988. A statistical and geometrical edge detector for sar images. IEEE Transactions on Geoscience and Remote Sensing 26(6), pp. 764–773.

Wright, P., Macklin, T., Willis, C. and Rye, T., 2005. Coherent Change Detection with SAR. In: European Radar Conference, EURAD 2005, Paris (France), pp. 17–20.

Ying, L., Demanet, L. and Candès, E. J., 2005. 3D Discrete Curvelet Transform. In: Proc. Wavelets XI conf. 2005, Symposium on Optical Sciende and Technology, San Diego (CA).

RAY TRACING AND SAR-TOMOGRAPHY FOR 3D ANALYSIS OF MICROWAVE SCATTERING AT MAN-MADE OBJECTS

S. Auer [a], X. Zhu [a], S. Hinz [b], R. Bamler [ac]

[a] Remote Sensing Technology, Technische Universität München, Arcisstrasse 21, 80333 München - (Stefan.Auer, Xiaoxiang.Zhu)@bv.tum.de

[b] Institute for Photogrammetry and Remote Sensing, Universität Karlsruhe, Kaiserstrasse 12, 76128 Karlsruhe - stefan.hinz@ipf.uni-karlsruhe.de

[c] Remote Sensing Technology Institute, German Aerospace Center (DLR), Münchner Strasse 20, 82234 Oberpfaffenhofen-Wessling - Richard.Bamler@dlr.de

Commission VI, WG VI/4

KEY WORDS: SAR Simulation, Ray Tracing, POV Ray, SAR Tomography, TerraSAR-X

ABSTRACT:

An inherent drawback of SAR imaging of complex 3D structures is the potential layover of more than one scatterer in one resolution cell. Such scatterers can be separated by tomographic processing of multiple SAR images acquired with different across-track baselines. Simulation tools may further support interpretation of such layover effects appearing in multi-body urban scenes. In this paper, an existing 2D simulation approach, developed for separating different kinds of reflection effects in the azimuth-range plane, is enhanced by including the elevation direction as third dimension and thus enabling the comparison of the SAR simulation results with 3D imaging techniques such as tomography. After introducing the simulation concept, tools for three-dimensional analysis of scattering effects are presented. Finally, simulated data are compared with real elevation data extracted from TerraSAR-X images for showing potential fields of application.

1. INTRODUCTION

High resolution SAR sensors like TerraSAR-X or Cosmo-SkyMed provide SAR images having a resolution of below one meter in spotlight mode. While in SAR images of coarse resolution several dominant scatterers from man-made objects at slightly different ranges may be condensed into a single pixel, these will be separable in high resolution images. Hence, more image features can be distinguished due to an increased number of deterministic effects and due to an increased signal to clutter ratio for dominant scatterers (Adam et al., 2008).

However, visual interpretation of image features in high resolution SAR images remains challenging due to range dependent geometrical effects. SAR maps the 3D world basically into a cylindrical coordinate system, where range and azimuth are the image coordinates and elevation is the coordinate, along which all scattering contributions are integrated, i.e. all scatterers are mapped into the same resolution cell in the azimuth-range plane if they have the same spatial distance with respect to the SAR sensor. Access to the third coordinate, elevation, is achieved by multi-baseline methods, like Persistent Scatterer Interferometry (Ferretti et al., 2001; Kampes, 2006) or SAR tomography (Reigber & Moreira, 2000; Fornaro et al, 2003; Zhu et al., 2008).

Simulation of scattering effects for urban areas may support visual interpretation of high resolution SAR images. In this context, Franceschetti and co-workers (Franceschetti et al., 1995) distinguish between two different kinds of SAR simulators: image simulators and SAR raw data simulators. In the past, different concepts have been presented for simulating artificial SAR images for urban areas (Balz, 2006; Mametsa et al., 2001) and for simulating SAR raw data by illuminating simplified building models (Franceschetti et al., 2003).

The simulator presented in this paper applies ray tracing algorithms and has been developed for simulating artificial SAR reflectivity maps (Auer et al., 2008). The approach is focused on geometrical correctness while physical effects and speckle effects are neglected. In addition to a reflectivity map containing all backscattered intensities, reflection effects are assigned to different image layers based on available bounce level information, i.e. separate layers for single bounce, double bounce, etc. Hence, interpretation of deterministic reflection phenomena appearing at man-made objects is simplified.

So far, for providing image data in azimuth and range, two out of three dimensions of the imaging system have been exploited. The novelty of the presented approach compared to other simulation concepts relates to the fact that the complete 3D geometry of the SAR imaging process is simulated and stored. This enables one, based on the simulated 2D SAR, to retrieve information about the existence of multiple scatterers in one resolution cell in the SAR image. We show that simulation of the distribution of point scatterers in elevation direction may support the interpretation of estimated elevation coordinates derived by SAR Tomography.

The structure of the paper is organized as follows. Firstly, the basic simulation concept is introduced in Section 2 where four major parts of the simulation concept are explained including necessary developments for extraction and analysis of elevation data. Simulation results displaying elevation data are compared with real data extracted from a TerraSAR-X image in Section 3. Finally, in Section 4, a short summary is given and future work is addressed.

2. SIMULATION CONCEPT

The simulation approach presented in this paper is based on ray tracing algorithms provided by POV Ray (Persistence of Vision Ray Tracer), a free-ware ray tracing software. Main advantages of POV Ray are free access to its source code, optimized processing time, separability of multiple reflections and existing interfaces to common 3D model formats. In order to provide necessary output data for two-dimensional analysis of reflection phenomena, additional parts have been included to POV Ray's source code. The simulation concept consists of four major parts:

- Modeling of scene objects (Section 2.1)
- Sampling of the 3D model scene in POV Ray (Section 2.2)
- Creation of reflectivity maps (Section 2.3)
- 3D analysis of reflection effects by means of output data provided by POV Ray (Section 2.4)

In the following subsections, the processing chain will be explained in more detail.

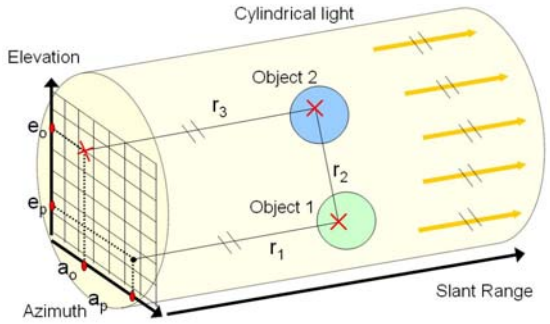


Figure 1: Approximation of SAR system by a cylindrical light source and an orthographic camera; 3D sampling due to coordinates in azimuth, slant-range, and elevation

2.1 Modeling of scene objects

First, the 3D scene to be illuminated by the virtual SAR sensor has to be described in the modeling step. 3D models can be designed in POV Ray or can be imported into the POV Ray environment. Then, parameters are adapted for describing the reflection behavior at object surfaces. To this end, POV Ray offers parametric models for specular reflection and diffuse reflection. A reflectivity factor for each surface defines the loss of intensity affecting rays specularly reflected at object surfaces.
In the case of a modeled SAR system both the light source and the camera are located at the same position in space. The concept for approximating the imaging geometry of the SAR system is shown in Figure 1. Focusing effects due to SAR processing in azimuth and range are considered by using a cylindrical light source and an orthographic camera whose image plane is hit perpendicularly by incoming signals.

2.2 Sampling of the 3D model scene

For analyzing backscattered signals within the modeled 3D scene, rays are followed in reverse direction starting at the center of an image pixel and ending at the ray's origin at the light source (Whitted, 1980). This concept is commonly referred to as Backwards Ray Tracing (Glassner, 2002). Since ray tracing is performed for each pixel of the image plane, output data for creating reflectivity maps is derived by discrete sampling of the three-dimensional object scene (Auer et al., 2008).
Coordinates in azimuth and range are derived by using depth information in slant-range provided during the sampling step. For instance, according to Figure 1, focused azimuth coordinates a_f and slant-range coordinates r_f of double bounce contributions are calculated by:

$$a_f = \frac{a_0 + a_p}{2} \quad (1)$$

$$r_f = \frac{r_1 + r_2 + r_3}{2} \quad (2)$$

where a_0, a_p = azimuth coordinates of the ray's origin and the ray's destination at the image plane

r_1, r_2, r_3 = depth values derived while tracing the ray through the 3D model scene

So far, only two axes of the three-dimensional imaging system - azimuth and range - have been used for reflection analysis (Auer et al., 2008). However, the third dimension, elevation, may provide potential to enhance the simulators capacities to 3D analysis of reflection effects. To this end, extraction of elevation data has been added to the sampling step. According to the imaging concept shown in Figure 1, the elevation coordinate for a double bounce contribution is derived by means of the following equation:

$$e_f = \frac{e_0 + e_p}{2} \quad (3)$$

where e_0, e_p = elevation coordinates of the ray's origin and the ray's destination at the image plane

At this point, elevation data derived during the sampling step shall be discussed in more detail. Due to Eq. (3) and the discrete sampling of the scene, all backscattering objects are assumed to behave as point scatterers. Resolution in elevation is not affected by limits occurring due to the size of sampling intervals along the elevation direction or the length of the elevation aperture (Nannini et al., 2008). From a physical point of view, deriving discrete points directly in elevation direction may be a disadvantage since comparison of the processed reflectivity function with a simulated one could be a desirable task. For instance, in the case of single bounce, the discrete concept will not be able to represent a planar surface continuously but only by discrete points.
For layover caused by multiple reflections along the elevation direction the discrete simulation concept is nonetheless reasonable since approaches for tomographic analysis also seek for scatterers whose backscattered intensity is concentrated in individual points along the elevation direction. Concentration on scene and SAR geometry and thereby neglecting the physical characteristics provides some advantages, though, to overcome well known limitations of tomographic analysis (Zhu et al., 2008). For instance, it leads to a better understanding of the SAR geometry in the elevation direction by means of simulating the reflectivity slice which is helpful for 3D reconstruction. Additionally, it has the potential to provide the number of scatterers in a cell as a priori for parametric tomographic estimators if the scene geometry is available at a very detailed level, e.g. based on airborne LIDAR surface models.

Eventually, the simulation process provides the following output data for each reflection contribution detected in the 3D object scene:

- coordinates in azimuth, slant range, and elevation [units: meter]
- intensity data [dimensionless value between 0 and 1]
- bounce level information for every reflection contribution [1 for single bounce, 2 for double bounce, etc.]
- flags marking specular reflection effects [value 0 or 1]

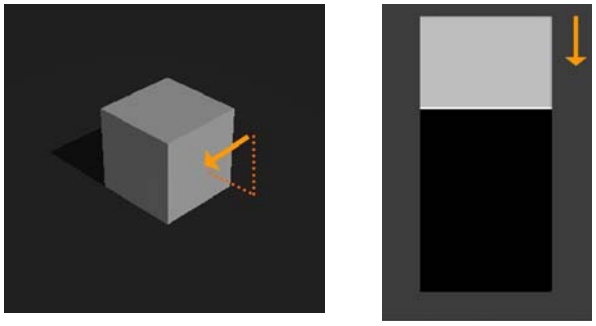


Figure 2: left: Simulation using box model having a size of 20 m x 20 m x 20 m, line of sight indicated by arrow; right: simulated reflectivity map simulated (slant-range indicated by arrow)

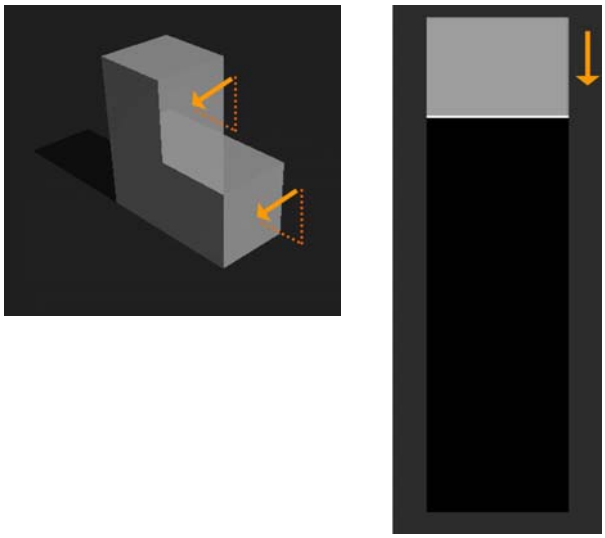


Figure 3: simulation using step model (left), line of sight indicated by arrow; simulated reflectivity map (right), slant-range indicated by arrow

2.3 Reflectivity maps in azimuth and slant range

Firstly, all reflection contributions are mapped into the azimuth – slant range plane. Afterwards, a regular grid is imposed onto the plane and intensity contributions are summed up for each image pixel. Figure 2 shows the resulting reflectivity map for a cube (dimensions: 20 m x 20 m x 20 m) which has been illuminated by the virtual SAR sensor using an incidence angle of 45 degrees. The size of one resolution cell has been fixed to cover 0.5 m x 0.5 m in azimuth and slant range. Surface parameters are chosen in a way that box surfaces can be clearly distinguished from ground parts, i.e. in the current example box surfaces show stronger diffuse backscattering than the surrounding ground. Following top-down in ground range direction, diffuse single bounce contributions of the ground are visible followed by a layover area of ground, wall of the box and top of the box. At the end of the layover area, a strong double bounce line is visible which is caused by the interaction between the front wall and the ground in front of the box.
For this type of scene geometry, a 2D simulation and analysis is usually sufficient. The next section will however illustrate examples that underline the necessity of including the elevation direction as third dimension into the simulation.

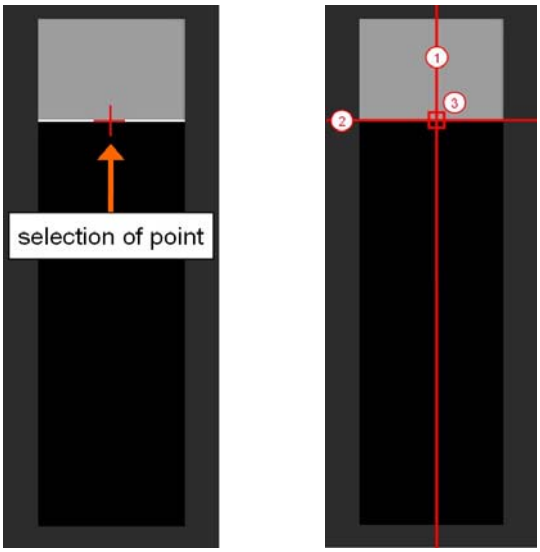


Figure 4: selection of pixel for elevation analysis (left); definition of three slices (right) in slant-range (1), azimuth (2), and elevation direction (3)

2.4 3D analysis of scattering effects

Figure 3 shows a reflectivity map simulated by illuminating a step model (width: 10 m, length 20 m, height 20 m). For providing the map, the same imaging geometry has been chosen as for the box example, i.e. the step was oriented in direction to the sensor and the incidence angle was fixed to 45 degrees in order to obtain specific overlay effects for single and double bounce contributions which are explained in the following. Compared to the reflectivity map containing the box model (Figure 2), the reflectivity map of the step shows similar characteristics. Both the layover area of single bounce contributions and the location of focused double bounce contributions are identical. Only the size of the shadow zone indicates a height difference between the illuminated objects. In the case of the step model, separation of dihedrals – two right angles at the steps – is impossible in the reflectivity map since all double bounce effects are condensed in one single line.
Hence, separation of scattering effects in elevation direction may be helpful since it enables to resolve layover effects for the purpose of distinguishing several scatterers within one resolution cell. To this end, an interactive click-tool has been included into the simulator for defining two-dimensional slices to be analyzed. In the case of the given reflectivity map for the step model, one pixel is selected, e.g. located in the double bounce area as shown in Figure 4. Based on the coordinates of the pixel center, three slices are defined:

- slice no. 1 for displaying elevation data in slant-range direction
- slice no. 2 for displaying elevation data in azimuth direction

- slice no. 3 for displaying intensities in elevation direction

According to the defined slices, necessary data in slant-range, azimuth and elevation are extracted out of the data pool provided by the sampling process in POV Ray.

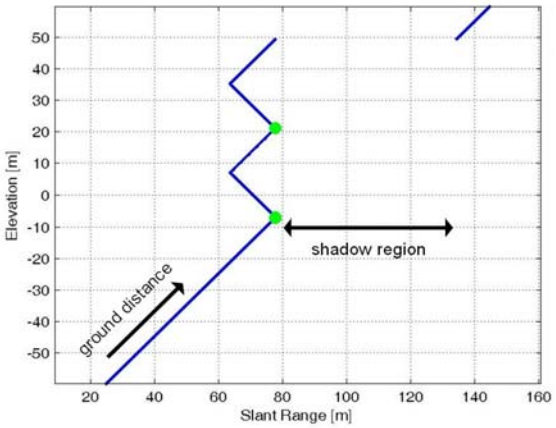


Figure 5: Slice 1: elevation heights in slant-range direction (slice 1 in Figure 4 corresponds to slant range interval 60 m to 140 m); blue: single bounce contributions, green: double bounce contributions

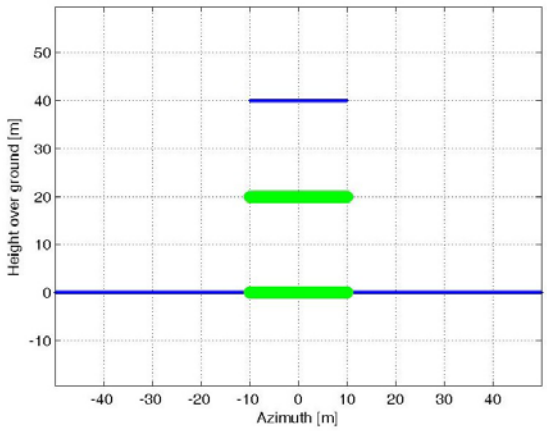


Figure 6: Slice 2: elevation information along azimuth direction displayed in height over ground; blue: single bounce contributions, green: double bounce contributions; zero level = level of ground surrounding the step

Since the incidence angle used for sampling the 3D model scene is known, slice no. 1 pointing in slant range direction can be presented by two versions, either by displaying elevation heights (Figure 5), i.e. elevation coordinates with respect to a master height situated in the center of the image plane used for sampling the scene, or by providing height information in height over ground geometry, i.e. heights with respect to the ground surrounding the box.
Following the slant-range direction from left to right, displaying height data in elevation heights enables to distinguish between range intervals containing one scatterer and areas containing several scatterers resulting in layover effects, which can not be separated in reflectivity maps such as shown in Fig. 3 (right). In Figure 5, reflection caused by direct backscattering are colored in blue color while double bounce contributions are indicated by green spots. Due to the incidence angle of 45 degrees, double bounce effects are focused at the same position in slant-range and are overlaid by both single bounce contributions at the ground and single bounce contributions reflected at the end of the step up-side.
Following slice no. 2 along its way, elevation information is shown along the azimuth direction in height over ground (Figure 6). After passing an interval of contributions directly backscattered at the ground, the layover region starts showing the width of the double bounce areas in azimuth, which are equal to the width of the step model. As expected, double bounce contributions caused by the interaction between perpendicular faces are concentrated at the corresponding intersection lines and, hence, show a height value of 0 and 20 meters, respectively.

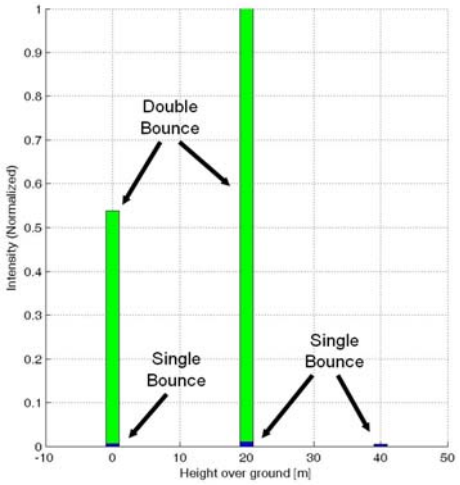


Figure 7: Slice 3: normalized intensities along elevation direction; step width in elevation: 2 meters; blue: single bounce contributions, green: double bounce contribution

Slice no. 3 pointing in elevation direction is shown in Figure 7. After the spatial sampling along elevation direction is chosen by the operator, intensity contributions are assigned to elevation intervals and summed up. Since the selected pixel is located within the double bounce area of two dihedrals, slice no. 3 shows two strong double bounce contributions caused by the interaction of step faces (colored in green) accompanied by weak direct backscattering derived at the step faces (colored in blue). Although the radiometric quality of detected intensity contributions is moderate due to simplified reflection models and the approximation of SAR signals by rays, proportions between single and double bounce intensities within one resolution cell are well represented.
In the following Section, simulation results will be compared to real data derived by tomographic analysis.

3. COMPARISON: SIMULATION VS. REAL DATA

For demonstrating potential applications of SAR simulation in elevation dimension, a practical example extracted from tomographic analysis using TerraSAR-X high resolution spotlight data is provided in this section and compared to simulation results.

3.1 Object modelling

Fig.8 shows the 2D intensity map for the convention center of Las Vegas acquired by TerraSAR-X. For the purpose of this paper, an azimuth-range pixel marked by a green dot has been taken as example. The complex valued measurement at this pixel corresponds to the integration of the reflected radar signal

along the elevation direction. It is located at the layover area. Fig.9 gives a closer look to the ground truth at that area. The left image shows the convention center visualized in Google Earth in which the pixel of interest is included in the area marked by a red block. The right image tells that the returns from the roof of the convention center and from the plaza near ground mainly contribute to the measurement of this pixel.

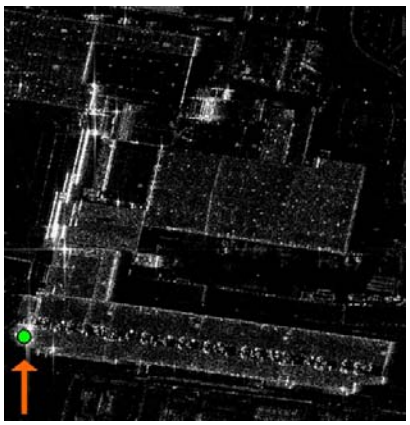


Figure 8: TerraSAR-X intensity image of convention center, Las Vegas; selected pixel marked by green spot

Figure 9: corresponding aerial image, © Google Earth

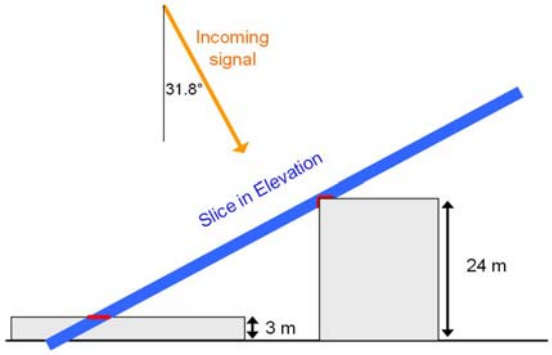


Figure 10: Model of the ground truth for the pixel of interest

For simulation, we simplify the ground truth to the following model as depicted in Fig. 10. The incidence angle for the acquisition is adapted to the real SAR acquisition and, hence, is 31.8 degrees. The taller building refers to the convention center which has a height of about 24m, while the lower building stands for the plaza near ground. Heights are measured over ground. The measurement for the pixel of interest refers to the integral of the returns from the objects included in the strip highlighted in blue color. For simulation purposes, two box models are used for modeling both the plaza and the convention center (Figure 11). The roughness of the plaza's surface is assumed to be slightly higher than the roughness of walls and roof parts of the convention center.

3.2 Simulation vs. real data

The reflectivity profile of the resolution cell along elevation direction is derived using the simulation concept described in Section 2. Pixel selection is adapted to extracted real data as shown in Figure 8. Afterwards, the resulting slice in elevation is displayed in height over ground geometry (Figure 12). Heights of reflecting objects are reliably extracted as two single bounce contributions at heights of 3 and 24 meters.

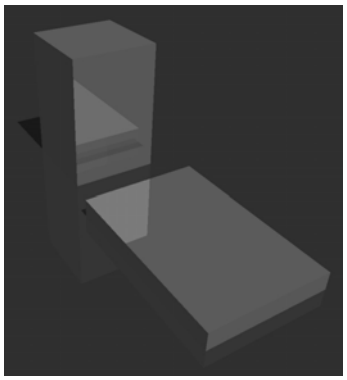


Figure 11: 3D model scene containing two boxes for approximating layover effect (flat box: 20 m x 30 m x 3 m; tall box: 15 m x 15 m x 24 m); diffuse backscattering behaviour at all box surfaces

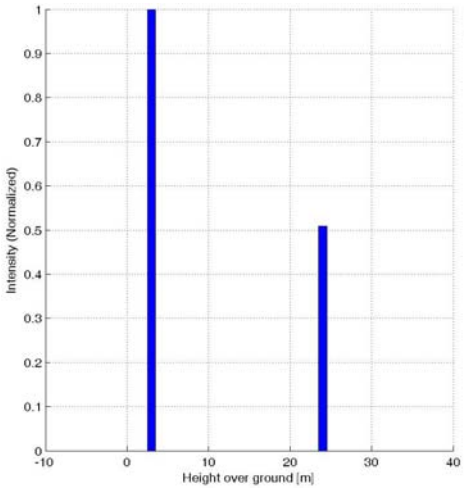


Figure 12: discrete elevation coordinates for backscattering objects; step-width in elevation: 1 meter

Fig. 13 shows the result of tomographic analysis for the corresponding position in real TerraSAR-X data. The reflection profile has been calculated with the approach described in (Zhu et al., 2008). As input data, 16 TerraSAR-X spotlight images with an across-track baseline range of 270m have been used. The peaks in reflection profile show nice correspondence with the simulated results, which underlines the accurate geometric properties of the simulation. However, it has to be noted that accurate estimation of intensity proportions is not possible as ground truth for surface properties was not available. At this point, simulated intensity values only indicate a stronger diffuse backscattering from the plaza which is also visible in the reflectivity map extracted from real SAR data. Enhanced information about the scattering behaviour of the plaza and the convention center may enable better simulation results in the

future. To not only compare the position of the peaks but also the shape of the reflection profile, the elevations of the simulated point scatterers (see Fig.12) were fed into the tomographic analysis assuming the same imaging configuration as for the real TerraSAR-X data. As can be seen from Fig. 14, the profile matches very well with the tomographic results from real data (Fig. 13). This example provides a validation of the SAR simulator in the third dimension by comparing to the tomographic analysis result using the TerraSAR-X data. In a further step, it can also be used for validation of tomographic algorithms by simulating the complex valued measurements of a data stack with different baseline distributions.

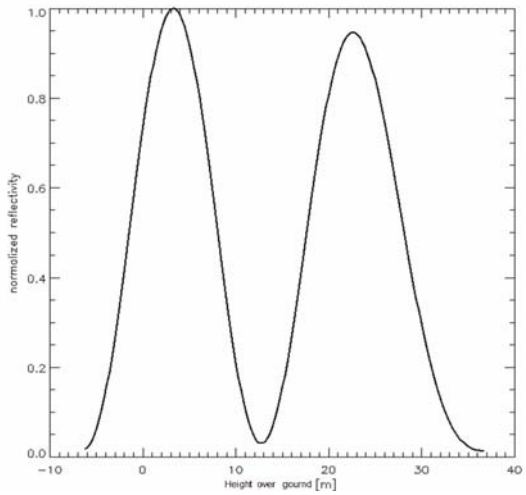


Figure 13: reflectivity function extracted from TerraSAR-X data by SAR-Tomography; intensity peaks estimated at heights of 3 m and 22.5 m

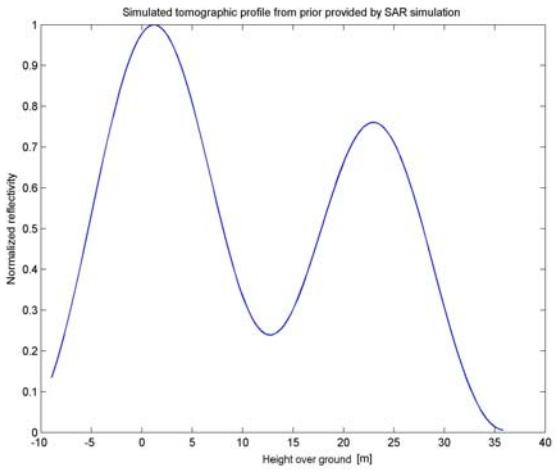


Figure 14: reflectivity function estimated from simulated data extracted from simulator by SAR-Tomography.

4. SUMMARY AND OUTLOOK

In this paper, a concept and its prototype implementation for 3D analysis of reflection effects has been presented. 3D model scenes are sampled by ray tracing techniques for providing necessary output data in azimuth, slant-range and elevation. Elevation slices are determined by pixel selection in reflectivity maps in the azimuth-range plane. Comparison of simulated data with real SAR data for a selected urban scene provided promising results. Further studies will have to show whether simulated elevation data may also support the geometrical analysis of more complex 3D urban scenes since visual interpretation of the simulation results is expected to become more complicated due to the increased number of visible building features. Meanwhile, the SAR estimator will be extended for the purpose of validation of tomographic algorithms.

5. REFERENCES

Adam, N.; Eineder, M.; Yague-Martinez, N.; Bamler, R., 2008. High Resolution Interferometric Stacking with TerraSAR-X. *Proceedings of IGARSS 08*, Boston, USA

Auer, S.; Hinz, S.; Bamler, R., 2008. Ray Tracing for Simulating Reflection Phenomena in SAR Images. *Proceedings of IGARSS 08*, Boston, USA

Balz, T. 2006. Real-Time SAR Simulation of Complex Scenes Using Programmable Graphics Processing Units. *Proceedings of the ISPRS TCVII Mid-term Symposium*

Ferretti, A.; Prati, C.; Rocca, F, 2001. Permanent scatterers in SAR interferometry. *IEEE Transactions on Geoscience and Remote Sensing, 39*, 8-20

Fornaro, G.; Serafino, F.; Soldovieri, F., 2003. Three-dimensional focusing with multipass SAR data. *IEEE Transactions on Geoscience and Remote Sensing, 41*, 507-517

Franceschetti, G.; Iodice, A.; Riccio, D.; Ruello, G., 2003. SAR raw signal simulation for urban structures. *IEEE Transactions on Geoscience and Remote Sensing, 41*, 1986-1995

Franceschetti, G.; Migliaccio, M.; Riccio, D., 1995. The SAR simulation: an overview. *Geoscience and Remote Sensing Symposium, 1995. IGARSS '95. 'Quantitative Remote Sensing for Science and Applications', International, 3*, 2283-2285 vol.3

Glassner, A. S., 2002. *An Introduction to Ray Tracing*. Morgan Kaufmann, 329

Kampes, B. M., 2006. *Radar Interferometry - Persistent Scatterer Technique*. Springer, 213

Mametsa, H. J.; Rouas, F.; Berges, A.; Latger, J, 2001. Imaging Radar simulation in realistic environment using shooting and bouncing rays technique. *Proceedings of SPIE: 4543. SAR Image Analysis, Modeling and Techniques IV, Toulouse*

Nannini, M.; Scheiber, R.; Moreira, A., 2008. On the Minimum Number of Tracks for SAR Tomography. *Proceedings of IGARSS 08*, Boston, USA

Reigber, A.; Moreira, A., 2000. First demonstration of airborne SAR tomography using multibaseline L-band data. *IEEE Transactions on Geoscience and Remote Sensing, 38*, 2142-2152

Whitted, T., 1980. An improved illumination model for shaded display. *Commun. ACM, 23*, 343-349

Zhu, X.; Adam, N.; Bamler, R., 2008. First Demonstration of Space-borne High Resolution SAR Tomography in Urban Environment Using TerraSAR-X Data. *Proceedings of CEOS SAR Workshop on Calibration and Validation*

THEORETICAL ANALYSIS OF BUILDING HEIGHT ESTIMATION USING SPACE-BORNE SAR-INTERFEROMETRY FOR RAPID MAPPING APPLICATIONS

Stefan Hinz[1], Sarah Abelen[2]

[1]Institute of Photogrammetry and Remote Sensing, Univeristät Karlsruhe, Kaiserstr. 12, 76 128 Karlsruhe
[2]Remote Sensing Technology, Technische Universität München, Arcisstr. 21, 81 333 München
Stefan.Hinz@ipf.uni-karlsruhe.de

Commission III, WG 5

KEY WORDS: Space-borne Interferometric SAR, Building Heights, TanDEM-X, CosmoSkymed

ABSTRACT:
The great potential of space-borne SAR images for semi- or fully-automatic mapping of topographic features has been shown by many approaches. While most of them focus on 2D mapping of topographic features, some preliminary research on the complex task of automatic delineation of 3D information in urban environments has been initiated in recent years. In this paper, we analyze the capabilities of new space-borne interferometric SAR missions – in particular the German TanDEM-X mission – with respect to their potential of deriving building heights. To this end, we summarize the mathematical framework and carry out a thorough analytical accuracy analysis involving various sensor and scene parameters.

1. INTRODUCTION

The new class of space-borne high resolution SAR sensors such as TerraSAR-X, SAR-Lupe or Cosmo-SkyMed is able to provide SAR images of 1-3m spatial resolution or even below in special spotlight modes. Naturally, the development of methods to automatically derive detailed cartographic information from this kind of data is a major issue driven by these missions. Since SAR is largely independent from illumination and weather conditions, it is furthermore an attractive imaging technology for acquiring area-wide information of regions hit by disasters such as floodings, landslides, or earthquakes.

The great potential of space-borne SAR images for semi- or fully-automatic mapping of 2D topographic features has been shown by many encouraging approaches, e.g., (Negri et al., 2006; Frey & Butenuth, 2009) for delineation of roads and (Jäger et al. 2007; Hänsch & Hellwich, 2008) for classification of agricultural features, just to name few recent ones. The derivation of 3D features is however more difficult, since these current civilian space-borne systems have only limited interferometric capabilities. While the acquisition of along-track interferometric image pairs is possible by programming special RADAR imaging modes (e.g. DRA mode or Aperture Switching mode for TerraSAR-X (Runge et al., 2006)) enabling the detection of moving objects (Suchandt et al., 2008; Weihing et al., 2008), none of the current civilian space systems is equipped with an across-track interferometer, which would provide the basis for deriving topographic heights (Bamler & Hartl, 1998; Cumming & Wong, 2005). The necessary across-track baseline is only given when forming an interferogram of two SAR acquisitions taken from the same orbit yet at different passes of the satellite, and thereby relying on the positional variation of the orbits. It is clear that the resulting interferograms suffer from decorrelation depending on temporal variability of the objects under investigation.

This situation will change once TerraSAR-X will be accompaigned by a second, quasi-identical SAR satellite in late 2009, leading to the TanDEM-X mission. Both satellites will fly almost in parallel forming a helix-like orbit pair (Zink et al. 2006). This configuration allows to acquiring SAR image pairs with variable across-track geometry resulting in a significantly improved interferometric coherence. The great benefit of single-pass across-track SAR interferometers has been intensively studied in the context of the Shuttle Radar Topography Mission (SRTM). Despite of the limited spatial resolution of SRTM data (approx. 25m), it was possible to compute a global digital elevation model with standardized height accuracy of few meters, see, e.g., the comprehensive overview given in (Rabus et al., 2003).

TanDEM-X will deliver high coherence interferometric data of the meter class. Although the mission is mainly designed to generate accurate digital elevation models satisfying HRTI-3 standards (Zink et al., 2006), it can be expected that this kind of data opens up a much wider field for specialized methods for 3D mapping of topographic features. The automated derivation of building heights or even the detailed reconstruction of buildings is certainly an important application amongst these.

Apart from the improved spatial resolution, a major difference between TanDEM-X and SRTM is the variable across-track baseline of TanDEM-X, whereas the baseline of SRTM was held quasi-constant due to the second antenna mounted at a 60m boom (and neglecting periodic baseline variations as consequence of thrusting). Hence, a thorough analysis of accuracy aspects of height estimation under the given flexibility of TanDEM-X is a key issue.

Following questions should be answered by the analysis:

- Which accuracy level in terms of building height estimation can be reached with interferometric data as it will be provided by TanDEM-X?
- Is this accuracy sufficient to derive object specific information for rapid mapping in the context of crisis management? Such information may comprise, e.g.,
    - the number of floors to estimate the amount of people living in a house
    - attached building parts of different height
    - the roof type (flat roof, saddle roof, etc.)
- How would the accuracy improve, if external data from GIS is included (e.g. digital building footprints)?

The reminder of this paper is organized as follows: Section 2 gives a brief review of state-of-the-art methods for delineating the 3D geometry of buildings from SAR images in general, eventually leading to a discussion of the boundary conditions of this study. The theoretical background for height estimation from across-track interferometry as well as error sources are compiled in Section 3, before Section 4 analyses the accuracy potential of deriving building heights under various given prerequisites. Finally, Section 5 draws conclusions in the light of the TanDEM-X mission and the results of this study.

2. 3D BUILDING GEOMETRY FROM SAR IMAGES

2.1. Overview

Over the past decades, a large variety of approaches for deriving 3D building information from SAR images has been developed. According to underlying methods and used data the different methods can be roughly grouped into following categories:

(a) height-from-shadow using mono- or multi-aspect data
(b) fitting prismatic models based on statistical optimization
(c) model-driven segmentation of pre-computed height data
(d) height estimation supported by feature detection / matching
(e) Exploiting layover areas in single or multiple InSAR pairs

To keep the overview focused, we only refer to the original work of each of these groups. We are aware that numerous approaches have been developed meanwhile, which could be assigned to one or more of these groups.

Ad a) Due to the oblique imaging geometry of SAR systems, buildings cause the well-known RADAR shadow, which basically corresponds to the occluded area at ground. As for conventional optical shape-from-shadow approaches, it only needs simple trigonometry to calculate the object height from the shadow boundary when knowing the sensor imaging geometry and assuming horizontal ground (similar for the layover area (Tupin, 2003)). A compilation of the corresponding formulae can be found, for instance, in (Sörgel et al. 2006). It is usually assumed that a shadow edge corresponds to a certain object edge, whose height is to be estimated. As only a few number of building edges can be matched to shadow edges for a specific viewing direction of the SAR, (Bolter & Leberl, 2000; Leberl & Bolter, 2001) generalize this approach to multi-aspect SAR and embed it into an iterative height estimation framework supported by InSAR cues. By this, building footprint and height are estimated simultaneously, yielding an accuracy of 1.5m – 2m for airborne SAR.

Ad b) The concept described in (Quartulli & Datcu, 2001; 2003) models the geometry of buildings and geometric relations between adjacent buildings by a number of parameters (position, length, width, height, roof slope, distance etc.). After initialization of model instances in image space, the parameters are statistically optimized using amplitude, coherence and interferometric phase information from the images. While this kind of thorough object-oriented modeling helps to cope with heavy noise and image derogations, it limits the approach to a small number of building shapes, not speak about the computational complexity mandatory for parameter optimization. This might one of the reasons why the results cannot prove the general feasibility of the approach and no accuracy analysis has been carried out; whereas, the mathematical formulation is very elegant.

Ad c) A purely data-driven strategy that complements the aforementioned approach is presented in (Gamba & Houshmand, 1999; Gamba et al., 2000). The procedure starts with the computation of the interferogram and derives level lines by segmenting it into height intervals. Level lines fulfilling certain shape constraints are selected as seed points to start a regiongrowing algorithm. This algorithm continues as long as segments can be added without exceeding a predefined threshold for co-planarity. The achieved accuracy using airborne C-band data is reported to be 2.5m for large industrial buildings. This method is in principle independent of the data source and can be applied to any kind of height models, as so for LIDAR-based height models (Gamba & Houshmand, 2000).

Ad d) While the former extraction strategy infers the semantics of buildings purely based on the roof geometry, approaches following the spirit of (Sörgel et al., 2003; Tison et al., 2007) include hypotheses of buildings, building parts, and/or adjacent context objects (roads, vegetation, etc.) from the very beginning of processing. To this end, a supervised classification and/or feature detection is carried out before building reconstruction. This may contain areal objects but also linear features and spots indicating double bounces at building walls, which become especially prominent in high resolution SAR (Stilla, 2007). The cues provided by these hypotheses are then iteratively grouped and optimized together with the heights derived from InSAR data until reasonably shaped buildings are extracted or hypotheses are rejected. Due to generic processing of multiple cues, this concept is easily extended to multi-aspect SAR data. The reported accuracy yields again 2 – 3m for the airborne case.

Ad e) The final group of approaches does not only include image features derived from SAR or InSAR data but models the complete interferometric phase profile for building walls and roofs (Thiele et al., 2007). Since vertical walls form layover areas as consequence of the oblique RADAR distance measurement, this kind of modelling implicitly contains the assumption that the main contribution of scattering in such layover areas is induced by building walls and not by clutter in front of the building or by the overlayed part of the roof. This approach can be generalized to SAR tomography (Reigber & Moreira, 2000; Fornaro et al., 2003) if more than one interferometric pair of the same viewing direction is available. (Zhu et al., 2008; 2009) show that deriving 3D information via tomographic analysis and statistical model selection can be adapted to pixelwise calculation of dense height maps of urban areas, thus linking the concepts of SAR tomography with Persistent Scatterer Interferometry (Ferretti et al., 2001; Kampes, 2006). These approaches are however in a preliminary stage so that a thorough accuracy analysis is not yet available.

2.2 Discussion

While each of the approaches is characterized by individual advantages and limitations, the latter category seems to be a good compromise between a data-driven strategy and object-oriented modeling. It is flexible in the sense that it is not restricted a-priori to specific building shapes. On the other hand, there are still object-oriented aspects included since typical building regularities are to identify in the InSAR data.

Concerning the utilization of shadow and layover effects one has to keep in mind that, especially in urban areas, layover appears very often and may also cover shadow from neighboring buildings. Hence, shadow areas are usually hard to

identify automatically, while layover areas still carry useful information, even if this is embedded in clutter (see the example of a repeat-pass TerraSAR-X interferogram in Figure 1). Moreover, the incorporation of additional knowledge about buildings may help to separate useful signal from clutter, especially as buildings have regular shapes and, often, digital maps indicating the building footprints are available.

Multi-baseline and multi-aspect approaches show great potential to reconstruct buildings with high accuracy and level of detail. However, the time needed to acquire the necessary images is usually too long for rapid mapping, especially in the context of providing crisis information. Consequently, the analysis in Section 3 concentrates on accuracy aspects of single-pass interferometry. The building heights are expected to be computed from the interferometric signal of layover regions. The inherent contribution of clutter in these areas is accommodated by some loss of interferometric coherence, which is also taken into account for the final height accuracy.

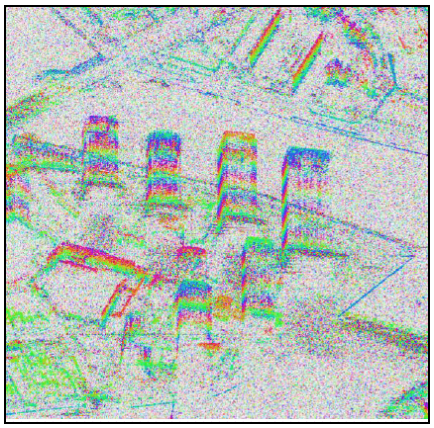


Figure 1: Interferometric fringes in layover area of tall buildings computed from dual-pass TerraSAR-X interferogram (courtesy M.Eineder, DLR).

3. HEIGHT ACCURACY OF INSAR MEASURMENTS – THEORY

In this section we revise the mathematical theory for relating specific sensor and scene parameters with the desired height accuracy for the case of space-borne SAR. A detailed derivation of the formulae can be found in (Bamler & Schättler, 1993; Bamler & Hartl, 1998; Cumming & Wong, 2005). Figure 2 (left) depicts the typical geometric configuration of across-track interferometry. The phase values of the two acquisitions can be derived from the well-known two-way range equation

$$\phi_1 = -\frac{2\pi}{\lambda} 2R + \phi_{scatt,1} \tag{1}$$

$$\phi_2 = -\frac{2\pi}{\lambda} 2\left(R + \Delta R\right) + \phi_{scatt,2} \tag{2}$$

where ϕ_1 and ϕ_2 are the SAR phases at a certain pixel, λ is the wavelength, R is the range between one antenna and the point on ground in viewing direction θ, and ΔR is the range difference induced by the baseline vector B and its component perpendicular to the viewing direction $B_\perp$, respectively. Under the assumption that the unknown phase contributions caused by random scattering $\phi_{scatt,1}$ and $\phi_{scatt,2}$ are identical

$$\phi_{scatt,1} \equiv \phi_{scatt,2} \tag{3}$$

one can express the interferometric phase ϕ for a certain point by

$$\phi = \phi_1 - \phi_2 = \frac{4\pi}{\lambda} \Delta R \tag{4}$$

In order to convert the this phase into height values z, it is useful to first formulate the functional relationship between ΔR and the direction perpendicular to R on ground, ζ (see Figure 2 (right)):

$$\zeta \cong -\frac{R_S}{B_\perp} \Delta R = -\frac{R_S}{B_\perp} \frac{\lambda}{4\pi} \phi \tag{5}$$

Solving for ϕ and projecting into the vertical direction z yields

$$\phi \cong -\frac{4\pi}{\lambda} \frac{B_\perp}{R} \frac{z}{\sin\theta} \tag{6}$$

Equation (6) is the basis to calculate the so-called phase-to-height sensitivity:

$$\frac{\partial\phi}{\partial z} = -\frac{4\pi}{\lambda} \frac{B_\perp}{R\sin\theta} \tag{7}$$

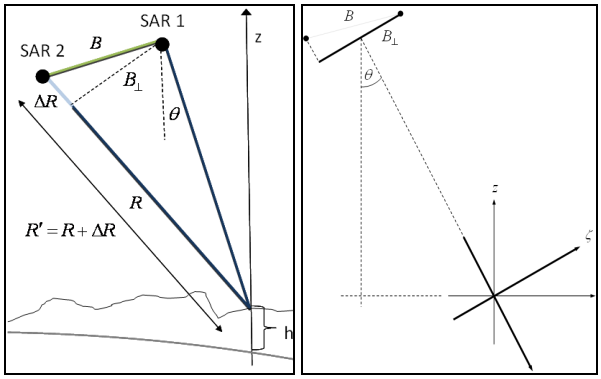


Figure 2: Geometric layout of across-track interferometry (left) and definition of local co-ordinate system on ground, ζ, (right).

Figure 3 illustrates the influence of varying incidence angle and baseline length on the phase-to-height-sensitivity. As can be seen, the interferometric measurement gets more and more sensitive the longer the baseline and the smaller (steeper) the incidence angle is.

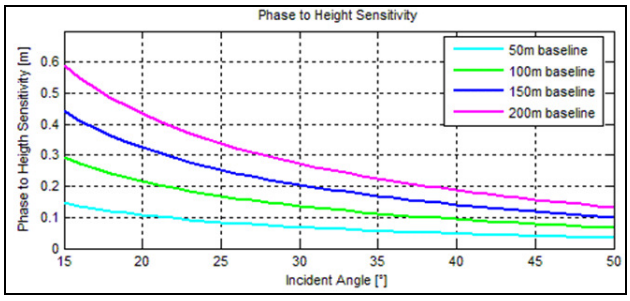


Figure 3: Influence of incidence angle and baseline on phase-to-height-sensitivity

In practice, the identity of Eq. (3) does not hold strictly, neither for ideal single-pass interferometers. Reasons therefore are varying bi-directional scattering, varying volume scattering, thermal noise, etc. However, a guess about the similarity of the two complex-valued SAR images u_1 and u_2 can be computed for each pixel $[i,k]$ by the coherence estimate $|\gamma| = |\hat{\gamma}[i,k]|$ calculated in a predefined local neighbourhood W:

$$|\gamma| = |\hat{\gamma}[i,k]| = \frac{\left|\sum_W u_1[i,k]\,u_2^*[i,k]\right|}{\sqrt{\sum_W |u_1[i,k]|^2 \sum_W |u_2[i,k]|^2}} \tag{8}$$

Based on the coherence estimate one can derive the probability density distribution (pdf) of the interferometric phase $\mathrm{pdf}(\phi; L)$ of the expectation $\bar{\phi}$ depending on the number of looks L, i.e. the amount of averaging independent pixels (Lee et al., 1994):

$$\mathrm{pdf}(\phi; L) = A(\phi; L) + B(\phi; L) \tag{9}$$

with

$$A(\phi; L) = \frac{\Gamma(L+1/2)\left(1-|\gamma|^2\right)^L |\gamma| \cos(\phi-\bar{\phi})}{2\sqrt{\pi}\;\Gamma(L)\left(1-|\gamma|^2\cos^2(\phi-\bar{\phi})\right)^{L+1/2}},$$

$$B(\phi; L) = \frac{\left(1-|\gamma|^2\right)^L}{2\pi}\;{}_2F_1\left(L, 1; 1/2; |\gamma|^2\cos^2(\phi-\bar{\phi})\right)$$

and $\Gamma(x) = (x-1)!$ being the Gamma function and ${}_2F_1(a,b;c;z)$ being the hypergeometric Gaussian function. Figure 4 shows the shape of Eq. 9 for a fixed coherence and varying averaging while, in Figure 5, averaging is fixed and coherence varies.

Although the pdf of the interferometric phase is not strictly Gaussian, it can be seen from the functions displayed in Figures 4 and 5 that the pdf's first- and second-order moment ($\bar{\phi}$ and σ_ϕ) carry the most information of this distribution. Furthermore, assuming that $\phi(z)$ is locally linear, one can write after Taylor expansion of $\phi(z)$ and omitting higher order terms:

$$\frac{\partial\phi}{\partial z} = \frac{\Delta\phi}{\Delta z} \tag{10}$$

Computing σ_ϕ numerically from Eq. 9 and inserting $\Delta\phi = \sigma_\phi$ and $\Delta z = \sigma_z$ into Eq. 10 yields finally the standard deviation of the height estimates:

$$\sigma_z = \frac{\sigma_\phi}{\partial\phi/\partial z} \tag{11}$$

Figure 6 visualizes the behaviour of the standard deviation of the interferometric phase for varying coherence and number of looks. An evident feature of this function is the large influence of averaging for moderate coherence values. Only four looks, for instance, improve the standard deviation approximately by 50% at a coherence of 0.65. Figure 7 shows typical height distributions for varying coherence and a specific fixed set of sensor parameters.

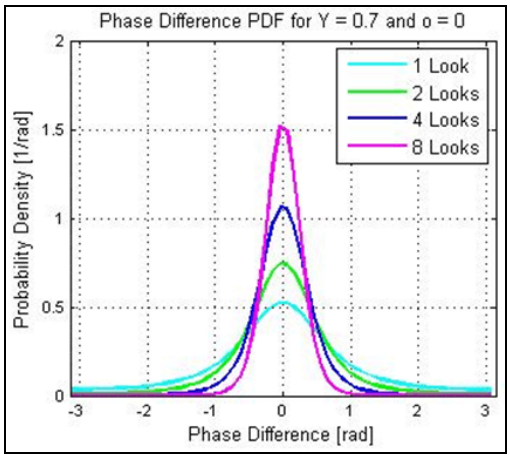


Figure 4: pdf of interferometric phase for fixed coherence and varying averaging.

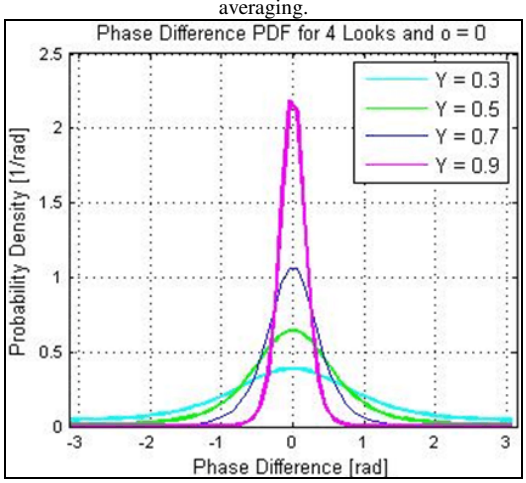


Figure 5: pdf of interferometric phase for fixed averaging and varying coherence.

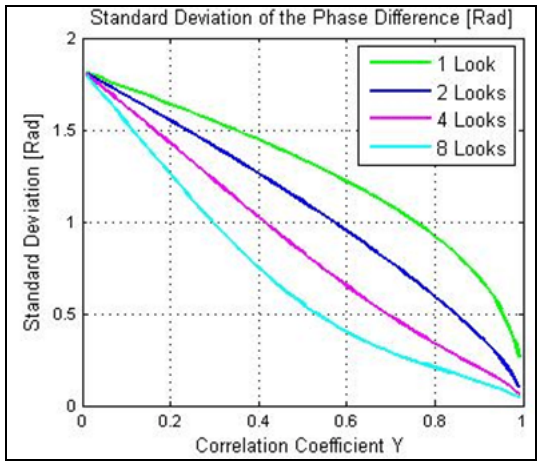


Figure 6: Standard deviation of interferometric phase for varying coherence and number of looks.

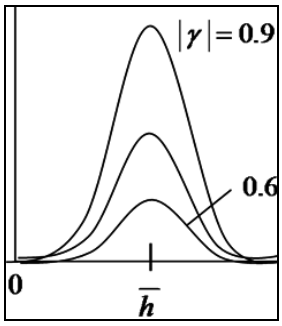


Figure 7: pdf of height for varying coherence (other parameters fixed).

4. ACCURACY OF BUILDING HEIGHT ESTIMATION

The accuracy assessment has been carried out with the scheme described above. Parameter values were chosen in accordance to the main TanDEM-X specifications. Mutual influence of following parameters has been investigated:

- baseline length
- incidence angle
- coherence
- amount of averaging/smoothing
- wavelength

Figure 8 exemplifies the specific case of moderate coherence (0.6) and only one single look, i.e. the original spatial resolution is maintained and no smoothing is done. A low coherence has been chosen by intention, although significantly better values are expected for TanDEM-X, since clutter may decrease the coherence in layover areas. The ideal case for this configuration suggests to selecting a long baseline (e.g. 150m) and a steep incidence angle (15°), which results in a height accuracy of approx. 2.5m. Incidence angles less close to the system limits would yield an accuracy of 5m – 7m.

The effect of improving coherence is illustrated in Figure 9, where height accuracy is plotted against coherence and various baselines. It can be seen that, for a baseline of 150m, height accuracy increases from 2.5m at a coherence of 0.6 up to 1.5m at a coherence of 0.9.

The final assessment deals with the influence of the number of looks. A coherence of 0.7 for the case of 200m baseline and 15° incidence angle enables the derivation of building heights with 1m accuracy for the given viewing geometry (see Figure 10). As this kind of averaging reduces the spatial resolution, it is reasonable to investigate the effect of smoothing onto the height accuracy only up to four looks. What is moreover evident from Figure 10 is that the increase of height accuracy is significant from one to two and three looks, but it is then gradually attenuating – especially for typical coherence values in the range of 0.6 – 0.8.

As a last remark we refer to the used wavelength (X-band in our calculations). Equation 7 shows that the RADAR wavelength is a constant factor for the phase-to-height-sensitivity, which directly propagates to the standard deviation of height measurements. A longer wavelength (L-band for instance) yields a worse phase-to-height sensitivity and consequently a worse height accuracy. It should be noted, however, that long wavelengths generally yield better interferometric coherence depending on the scene characteristics. Hence, this effect could be partly compensated.

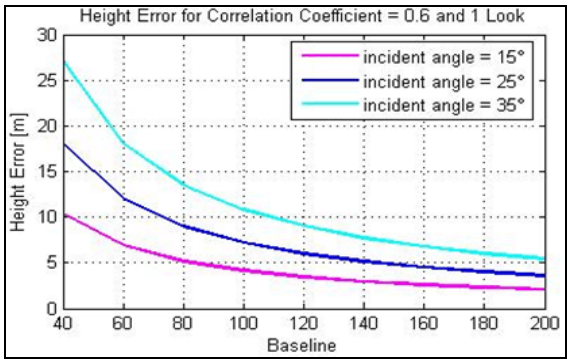


Figure 8: Height accuracy for varying baseline and incidence angle while other parameters are fixed.

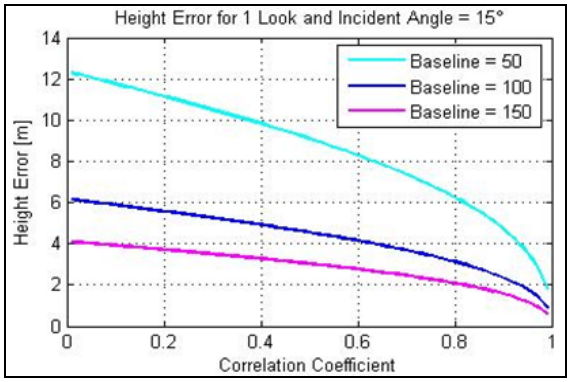


Figure 9: Height accuracy for varying baseline and coherence while other parameters are fixed.

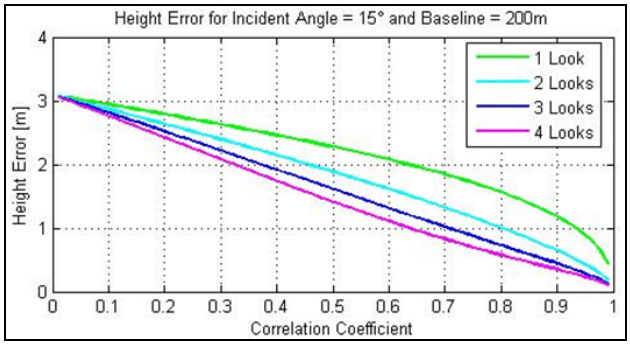


Figure 10: Height accuracy for varying looks and coherence while other parameters are fixed.

5. DISCUSSION AND CONCLUSION

The above analysis shows that space-borne interferometric SAR systems like TanDEM-X will allow to measure vertical heights with a standard deviation of roughly 1.5m, which also holds for the case of moderate coherence in layover areas. Regarding the application of rapid mapping, this accuracy will certainly allow the estimation of the number of floors or the detection of changes in the 3D building geometry. However, baselines and incidence angles have to be chosen carefully, as they are close to the technical limits (i.e. “critical baseline” and steep viewing angle). These constraints can be relaxed when additional data in form of digital ground plans is available. These allow, for instance, the utilization of specialized filters instead of simple multi-looking. The geometry of the filter mask can then be adapted to the respective building shape to include as much pixels as possible as observations into the height measurement. Recall Figure 6 to see how the standard deviation of interferometric phase improves with increasing number of observations.

Further research will include investigations of the validity of Eq. 10 in real scenarios, e.g. by calculating the accuracy of heights based on phase differences while the phase estimate for the ground may have better accuracy than those of walls and roof. Another line of research could be the use of explicit models for scatterers typically appearing at buildings, instead of modeling their influence only implicitly by a lower coherence.

REFERENCES

Bamler, R., Hartl, P., 1998. Synthetic aperture radar interferometry. Inverse Problems: pp. R1 – R54.

Bamler, R., Schättler, B., 1993. SAR geocoding. Wichmann, Karlsruhe, chapter 3, pp. 53 – 102.

Bolter, R., Leberl, F., 2000. Phenomenology-based and interferometry-guided building reconstruction from multiple SAR images, Proc. of EUSAR 2000, pp. 687–690.

Cumming, I., Wong, F., 2005. Digital Processing of Synthetic Aperture Radar Data, Artech House, Boston, 2005.

Ferretti, A., Prati, C., Rocca, F, 2001. Permanent scatterers in SAR interferometry. IEEE Transactions on Geoscience and Remote Sensing, 39, 8-20

Fornaro, G., Serafino, F., Soldovieri, F., 2003. Three-dimensional focusing with multipass SAR data. IEEE Transactions on Geoscience and Remote Sensing, 41, 507-517

Frey, D., Butenuth, M., 2009. Analysis of Road Networks after Natural Disasters using Multi-sensorial Remote Sensing Techniques. In: Eckhard, Seyfert (eds.), Publikationen der Deutschen Gesellschaft für Photogrammetrie, Fernerkundung und Geoinformation 18, in press.

Gamba. P., Houshmand, B., 1999. Three dimensional urban characterization by IFSAR measurements, Proc. Of IGARSS'99, pp. 2401–2403.

Gamba. P., Houshmand, B., 2000. Digital surface models and building extraction: A comparison of IFSAR and LIDAR data, IEEE Transactions on Geoscience and Remote Sensing 38(4): 1959–1968.

Gamba. P., Houshmand, B., Saccani, M., 2000. Detection and extraction of buildings from interferometric SAR data, IEEE Transactions on Geoscience and Remote Sensing 38(1): 611–617.

Hänsch, R., Hellwich, O. 2008. Weighted Pyramid Linking for Segmentation of Fully-Polarimetric SAR Data, International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences 37(/B7a), pp. 95-100

Jäger, M., Neumann, M., Guillaso, S., Reigber, A., 2007. A Self-Initializing PolInSAR Classifier Using Interferometric Phase Differences, IEEE Transactions on Geoscience and Remote Sensing 45(11): 3503-3518.

Kampes, B. M., 2006. Radar Interferometry – The Persistent Scatterer Technique. *Springer*-Verlag, Berlin, Germany.

Leberl, F., Bolter, R., 2001. Building Reconstruction from Synthetic Aperture Radar Images and Interferometry. In: Baltsavias, E., Grün, A., van Gool, L. (eds.): Automatic Extraction of Man-Made Objects from Aerial and Space Images (III). Balkema Publishers, Lisse, The Netherlands.

Lee, J-S., Hoppel, K., Mango, S., Miller, A., 1994. Intensity and phase statistics of multilook polarimetric and interferometric SAR imagery IEEE Transactions on Geoscience and Remote Sensing 32: 1017–28

Negri, M., Gamba, P., Lisini, G., Tupin, F., 2006. Junction-Aware Extraction and Regularization of Urban Road Networks in High Resolution SAR Images. *IEEE Transactions on Geoscience and Remote Sensing* 44(10): 2962-2971.

Quartulli, M., Dactu, M. 2001. Bayesian model based city reconstruction from high resolution ISAR data, IEEE/ISPRS Joint Workshop on Remote Sensing and Data over Urban Areas, on CD

Quartulli, M., Datcu, M., 2003. Stochastic modelling for structure reconstruction from high-resolution SAR data, Proc. of IGARSS'03, pp. 4080–4082.

Rabus, B., Eineder, M., Roth, A., Bamler, R., 2003. The Shuttle Radar Topography Mission (SRTM) – A New Class of Digital Elevation Models Acquired by Spaceborne Radar. ISPRS Journal of Photogrammetry & Remote Sensing 57(4): 241 - 262

Reigber, A., Moreira, A., 2000. First demonstration of airborne SAR tomography using multibaseline L-band data. IEEE Transactions on Geoscience and Remote Sensing, 38, 2142-2152

Runge, H., Laux, C., Metzig, R., Steinbrecher, U., 2006. Performance Analysis of Virtual Multi-Channel TS-X SAR-Modes. Proceedings of EUSAR'06, Dresden, Germany, on CD.

Sörgel, U., Thoennessen, U., Stilla, U., 2003. Iterative building reconstruction from multi-aspect InSAR data," ISPRS Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences, vol. 34.

Sörgel, U., Thoennessen, U., Brenner, A., Stilla, U., 2006. High-resolution SAR data: new opportunities and challenges for the analysis of urban areas. Radar, Sonar and Navigation, IEE Proceedings 153, 294-300

Stilla, U., 2007. High Resolution Radar Imaging of Urban Areas, Photogrammetric Week, Wichmann-Verlag, pp. 149–158

Suchandt, S., Runge, H., Breit, H., Kotenkov, A., Weihing, D., Hinz, S., Palubinskas, G., 2008. Traffic Measurements with TerraSAR-X: Processing System Overview and First Results. In: Proceedings of EUSAR '08, Friedrichshafen, Germany, on CD.

Thiele, A., Thoennessen, U., Cadario, E., Schulz, K., Soergel, U. 2007. Building Recognition from Multi-Aspect High Resolution InSAR Data in Urban Areas. In: IEEE Transactions on Geoscience and Remote Sensing 45 (11): 3583-3593.

Tison, C., Tupin, F., Maitre, H. 2007. A fusion scheme for joint retrieval of urban height map and classification from high resolution interferometric SAR images, IEEE Transactions on Geoscience and Remote Sensing 45(2): 495–505.

Tupin, F., 2003. Extraction of 3D information using overlay detection on SAR images, GRSS/ISPRS Joint Workshop on "Data Fusion and Remote Sensing over Urban Areas", pp. 72–76.

Weihing, D., Suchandt, S., Hinz, S., Runge, H., Bamler, R., 2008. Traffic Parameter Estimation Using TerraSAR-X Data. In: International Archives of Photogrammetry, Remote Sensing and Spatial Geoinformation Sciences, Vol 37(B7), pp. 153 – 156.

Zhu, X., Adam, N., Bamler, R., 2008. First Demonstration of Space-borne High Resolution SAR Tomography in Urban Environment Using TerraSAR-X Data. Proceedings of CEOS SAR Workshop on Calibration and Validation

Zhu, X., Adam, N., Bamler, R., 2009. Space-borne High Resolution SAR Tomography: Experiments in Urban Environment Using TerraSAR-X Data. Proc. of URBAN 2009, Shanghai, CN, to appear.

Zink, M., Fiedler, H., Hajnsek, I., Krieger, G., Moreira, A., Werner, M., 2006. The TanDEM-X Mission Concept. Proc. of IGARSS 2006, on CD

FUSION OF OPTICAL AND INSAR FEATURES FOR BUILDING RECOGNITION IN URBAN AREAS

J. D. Wegner [a, *], A. Thiele [b], U. Soergel [a]

[a] Institute of Photogrammetry and Geoinformation, Leibniz University Hannover, Hannover, Germany – (wegner, soergel)@ipi.uni-hannover.de

[b] FGAN-FOM Research Institute of Optronics and Pattern Recognition, Ettlingen, Germany – thiele@fom.fgan.de

Commission III, WG III/4

KEY WORDS: Remote Sensing, Fusion, Feature Extraction, InSAR Data, Optical Data, Urban Area

ABSTRACT:

State-of-the-art space borne SAR sensors are capable of acquiring imagery with a geometric resolution of one meter while airborne SAR systems provide even finer ground sampling distance. In such data, individual objects in urban areas like bridges and buildings become visible in detail. However, the side-looking sensor principle leads to occlusion and layover effects that hamper interpretability. As a consequence, SAR data is often analysed in combination with complementary data from topographic maps or optical remote sensing images. This work focuses on the combination of features from InSAR data and optical aerial imagery for building recognition in dense urban areas. It is shown that a combined analysis of InSAR and optical data very much improves detection results compared to building recognition based on merely a single data source.

1. INTRODUCTION

Due to its independence of daylight and all-weather capability, synthetic aperture radar (SAR) has become a key remote sensing technique in the last decades. One main application scenario arises in crisis situations when the acquisition of a scene is required immediately for rapid hazard response. Urban areas play a key-role since the lives of thousands of people may be in danger in a relatively small area. In SAR data of one meter geometric resolution collected by modern space borne sensors such as TerraSAR-X and Cosmo-SkyMed, the geometric extent of individual objects like bridges, buildings and roads becomes visible. In airborne data such objects are imaged with even more detail. However, shadowing and layover effects, typical for SAR image acquisitions in urban areas, complicate interpretation. Small buildings are often occluded by higher ones while façades overlap with trees and cars on the streets. In addition, the appearance of an individual building in the image highly depends on the sensor's aspect. Buildings that are not oriented in azimuth direction with respect to the sensor are often hard to detect. This drawback can be partly overcome by using SAR images from multiple aspects (Xu and Jin, 2007). Building recognition and reconstruction can be further improved based on interferometric SAR (InSAR) acquisitions from two orthogonal flight directions (Thiele et al., 2007).

Nevertheless, automatic urban scene analysis based on SAR data alone is hard to conduct. SAR data interpretation can be supported with additional information from GIS databases or high-resolution optical imagery. Optical images have the advantage of being widely available. In (Soergel et al., 2007) high-resolution airborne InSAR data is combined with an optical aerial image in order to three-dimensionally reconstruct bridges over water. Tupin and Roux (2003) propose an approach to automatically extract footprints of large flat-roofed buildings based on line features by means of a SAR amplitude image and an optical aerial image. Furthermore, homogeneous regions in an aerial photo, represented in a region adjacency graph, are used in (Tupin and Roux, 2005) to regularize elevation data derived from radargrammetric processing of a SAR image pair by means of Markov Random Fields.

In this paper, an approach for building recognition in dense urban areas is presented that combines line features from mono-aspect InSAR data with classification results from one optical aerial image. Building corner lines extracted from InSAR data are introduced as features into a classification framework that is based on a segmentation of the optical image. Optical features and InSAR lines are jointly used in order to evaluate building hypothesis. The focus is on the fusion approach of building primitive hypothesis.

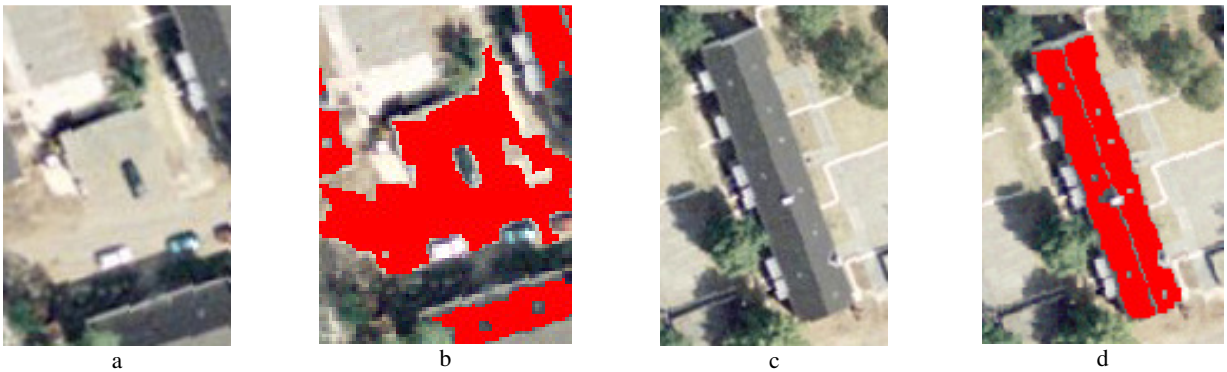
2. ANALYSIS OF OPTICAL DATA

Optical images provide high resolution multi-spectral information of urban scenes. For human interpreters they are by far more intuitive to understand than SAR data since the imaging geometry corresponds to the human eye. In aerial imagery of 0.3 meters resolution, like used in this project, building roofs become visible in great detail. In addition, façade details may appear in the image if high buildings situated far away from the nadir point of the sensor are imaged.

2.1 Appearance of Buildings

The appearance of an individual building mapped by any imaging sensor is both governed by its own properties (e.g., material, geometry) as well as by sensor characteristics (e.g., principle, spectral domain, pose), which have to be considered for recognition. For example, in optical images acquired from a near nadir perspective, building roofs are the most important features for automatic detection. Shadows are also good indicators for buildings (Fig. 1) and distinguish them, for instance, from road segments or parking lots. In western

* Corresponding author. This is useful to know for communication with the appropriate person in cases with more than one author.



a b c d

Figure 1. Flat-roofed (a) and gable-roofed (c) building in optical image overlaid with corresponding regions after segmentation (b,c)

countries rooftops look usually grey, reddish or brownish but almost never green. Roof types can roughly be subdivided into flat roofs and gable-roofs. Flat roofs coincide often with rather homogeneous image regions (Fig. 1a) while gable-roofs sometimes appear less homogeneous. Chimneys and shadows cast by chimneys may further complicate roof extraction if homogeneous planes are fit to roofs (Fig. 1 c,d). Due to similar colour of adjacent roof and street regions, such entities are sometimes hard to be told apart even for human interpreters (Fig. 1 a,b).

In this work the focus is on fusion of building primitive hypotheses delivered by approaches from the literature, tailored to the specific constraints that are determined by the particularities of the optical and microwave realm, respectively. With respect to the visible domain, a robust model-based roof detection approach introduced in Mueller and Zaum (2005), known to deliver good results, was used. It is based on an initial region growing step yielding homogeneous segments. As a consequence of the previously outlined diverse appearance of building roofs in optical imagery, such segmentation may sometimes lead to suboptimal results if contrast between roof regions and adjacent regions is very low (Fig. 1a). Thus, the region growing step can lead to erroneous roof segments (Fig. 1b). Gable-roofs usually split up into at least two segments if they are not oriented along the sun illumination direction (Fig. 1. c,d). Sometimes, gable-roofs may split up into even more than two segments and only parts are evaluated as roof regions. In such cases, the introduction of building hints from SAR data can highly improve building detection.

2.2 Feature Extraction

The building roof extraction approach consists of a low-level and a subsequent high-level image processing step (Mueller and Zaum, 2005). The low-level step includes transformation of the RGB image to HSI (Hue Saturation Intensity) representation, a segmentation of building hypotheses in the intensity image and the application of morphological operators in order to close small holes. Region growing, initialized with regularly distributed seed points on a grid, is used as image segmentation method. Seed points that fall into a grid cell which either consists of shadow or features a greenish hue value are erased and no region growing is conducted. Adjacent roof regions having a significant shadow region next to them are merged. This step is important for gable-roofed buildings because sometimes the roof is split at the roof ridge due to different illumination of the two roof parts. However, gable-roofs that were split up into more than two segments are not merged to one single segment which is the main reason for undetected buildings later-on in the process.

Features are extracted for each roof hypothesis in order to prepare for classification. Four different feature types are used, based on geometry, shape, radiometry, and structure. Geometric features are the region size and its perimeter. The shape of a building region is described by its compactness and length. Right angles, distinguishing roofs from trees in the real world, are not used as a shape feature since the region growing step may lead to segments that are not rectangular although they represent roofs (Fig. 1b). Radiometry is used in order to sort out regions with a high percentage of green pixels. Structural features are for example neighbouring building regions and shadows cast by the potential building. Shadows are good hints for elevated objects. In order to not take into account shadows cast by trees, only shadows with relatively straight borders are considered as belonging to buildings.

Finally, a classification based on the previously determined feature vector takes place (see chapter 4.2 for details). All necessary evaluation intervals and thresholds were learned from manually classified training regions.

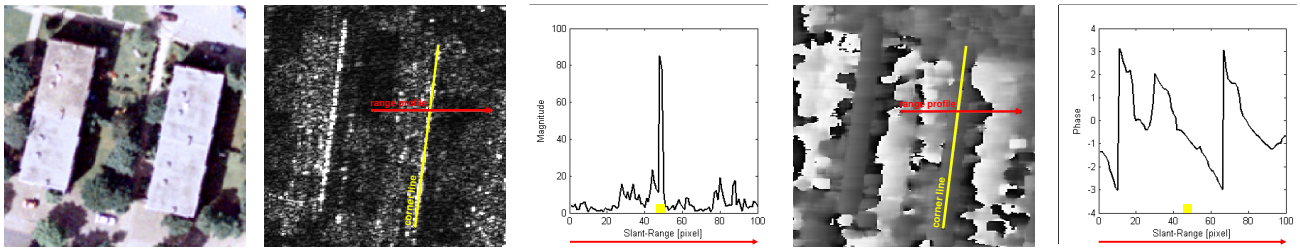
3. ANALYSIS OF INSAR DATA

3.1 Appearance of Buildings

The appearance of buildings in InSAR data is characterized by the oblique illumination of the scene and therefore the image projection in slant range geometry. Furthermore, it depends on sensor parameters, on properties of the imaged object itself, and on the object's direct environment.

In Fig. 2 an example of flat-roofed buildings in optical (Fig. 2a) and InSAR data (Fig. 2 b,d) is given. The appearance of different building types and effects that occur if the scene is illuminated from two orthogonal flight directions have been comprehensively discussed in Thiele et al. (2007 and 2008).

The magnitude profile of a building is typically a sequence of areas of various signal properties: layover, corner reflector between ground and building wall, roof, and finally radar shadow (Fig. 2c). The layover area is the building signal situated the closest to the sensor in the image because its distance is the shortest. It usually appears bright due to superposition of backscatter from ground, façade, and roof. The layover area ends at the bright so-called corner reflector line. This salient feature is caused by double-bounce reflection at a dihedral corner reflector spanned by ground and wall along the building. This line coincides with a part of the building footprint and can be distinguished from other lines of bright scattering using the InSAR phases (see Fig. 2d and profile in Fig. 2e). The single backscatter signal of the building roof is either included in the layover mixture or scattered away from



a b c d e

Figure 2. Appearance of flat-roofed buildings in optical data (a), in SAR magnitude data with illumination from right to left (b,c) and InSAR phase data (d,e)

the sensor depending on roof structure and illumination geometry. Ground behind the building is partly occluded by the building shadow leading to a dark region in the image.
A building also leads to specific patterns in the interferometric phase data (Fig. 2d and Fig. 2e) because the phase value of a single range cell results from a mixture of the backscatter of different contributors, such as ground, façade, and roof in the layover area. Again, the appearance is characterized by a layover region and a homogeneous roof region (in Fig. 2 not observable because of the narrow building width). The phase of the terrain enclosing the building is displayed slightly darker. A similar phase value is calculated at the building corner location, which is used for the detection of building footprints. Since no signal is received in shadow area, the related InSAR phase carries no useful signal but noise only.

3.2 Feature Extraction

This approach of building recognition in InSAR data is based on the detection of parts of the building footprint. First, the segmentation of bright lines is carried out in the magnitude data. Based on this set of lines, only the ones caused by a dihedral corner reflector spanned by ground and building wall are used as building hints. In order to exclude all lines that do not fulfil this criterion, the local InSAR heights are analysed. Finally, the filtered corner lines are projected into the same ground range geometry as the optical data.

3.2.1 Corner Line Segmentation

As previously discussed, the bright corner lines are very useful hints to buildings since they provide information about the true location of a part of the building footprint. The full process of corner line detection is shown in Fig. 3, upper row.
The line detection is carried out in slant range geometry based on the original magnitude images (Fig. 3 “Magnitude”) by using an adapted ratio line detector according to Tupin et al. (1998). This template detector determines the probability of a pixel of belonging to a line. In our case, eight different template orientations are considered. The probability image for the vertical template orientation is shown in Fig. 3 “Line”. Thereafter, line segments are assembled based on the eight probability images and their respective window orientation. The resulting segments are fitted to straight lines and edges, respectively, by linear approximation and subsequent prolongation (yellow lines in Fig. 3).

3.2.2 Geocoding of Building Features

After line extraction, the interferometric heights are calculated as described in (Thiele et al., 2007). Results are shown in Fig. 3 “Heights”. Local InSAR heights are investigated in order to discriminate lines caused by direct reflection and lines due to double-bounce reflection between either ground and wall or roof and substructures. For this filter step, the height difference between Digital Surface Model (DSM) and Digital Terrain Model (DTM) is used.
The DSM is given by the calculated InSAR heights. In order to derive the DTM from it, a filter mask is computed to define the DSM pixels which are considered in the DTM generation. Only pixels with a high coherence value (Fig. 3 “Coherence”) and an InSAR height close to the global mean terrain height are considered in equation 1 (Fig. 3 “Mask”).

$$x_{mask,i} = \begin{cases} 1, if\ \ x_{coh,i} \geq 0.5\ \ and\ \ \left(x_{h,i} - \overline{x_h}\right) \leq \pm\sigma_{\overline{x_h}} \\ 0, else \end{cases} \quad (1)$$

Based on this mask and the InSAR heights, a DTM height value is calculated over an area of 50 m x 50 m in ground range geometry (Fig. 3 “DTM”). Thereafter, the height differences (i.e., a normalized DSM) between DSM and DTM are calculated (Fig. 3 “Height difference”).
In the following line filtering step, lines are considered as real building corner lines if their neighbouring pixels show a low mean height difference value (Fig. 3 “Height difference”, rescaled for visualization). The filtered real corner lines are displayed in Fig. 3 (red lines). Final geo-coding of these corner lines is carried out using the InSAR heights. The resulting geographic position of the corner lines superimposed onto the optical image is displayed for the entire test site in Fig. 5b.

4. FUSION OF EXTRACTION OUTCOMES

In order to accurately combine features from InSAR data and the optical image, different sensor geometries and projections have to be considered carefully. It is required that both feature sets are projected to the same geometry, i.e., all data have to be transformed to a common coordinate system (Thiele et al., 2006). In addition, a fusion and classification framework for combining the detection outcomes from the optical image and from the InSAR data has to be set up.

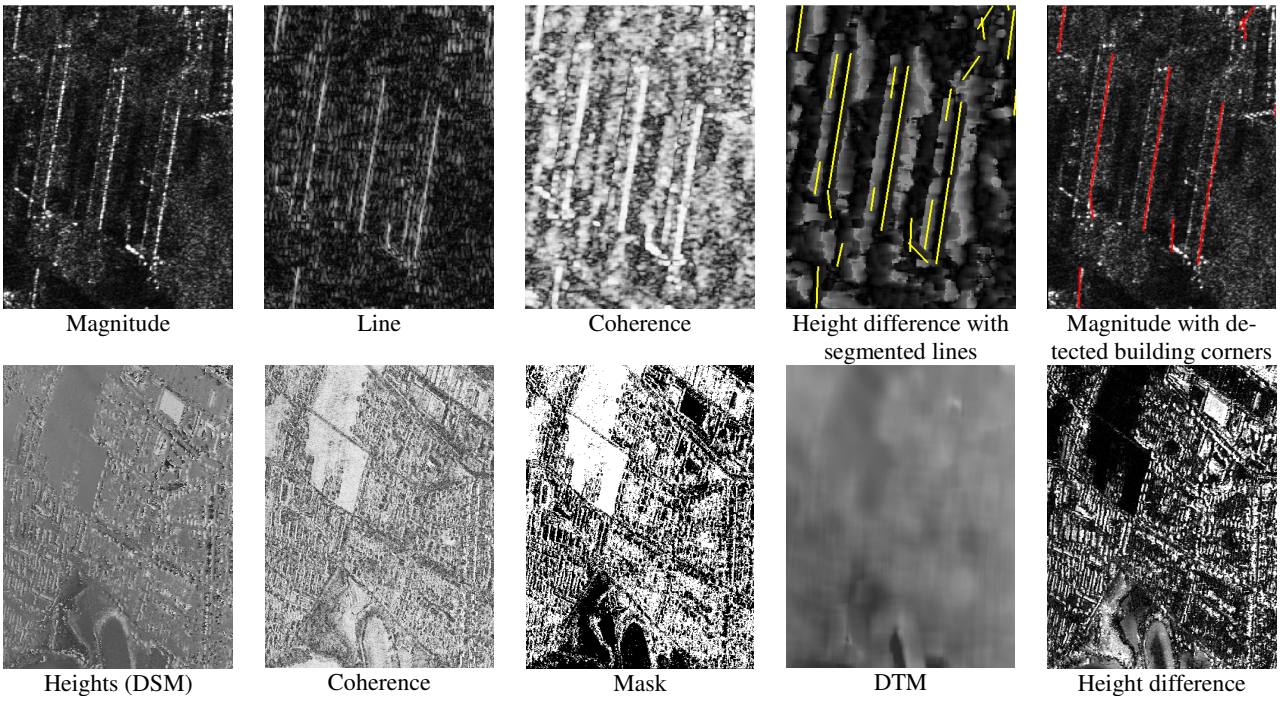


Figure 3. Upper row: steps of building corner segmentation in slant range geometry with illumination direction from left to right; lower row: steps of the InSAR height filtering and slant range to ground range projection of the building corner lines

4.1 Sensor geometries

The particularities of Synthetic Aperture Radar (SAR) and optical cameras in terms of sensor principle and viewing geometry result in very different properties of the observed objects in the acquired imagery. In Fig. 4a an elevated object *P* of height *h* above ground is imaged by both a SAR sensor and an optical sensor (OPT). SAR is an active technique measuring slant ranges to ground objects with a rather poor angular resolution in elevation direction. Layover, foreshortening, and shadowing effects consequently occur and complicate the interpretation of urban scenes. Buildings therefore are displaced towards the sensor. Point *P* in Fig. 4a is thus mapped to point *PS* in the image. The degree of displacement depends on the object height *h* and the off-nadir angle θ_1 of the SAR-sensor.

By contrast, optical sensors are passive sensors acquiring images with small off-nadir angles. No distances but angles to ground objects are measured. Elevated objects like *P* in Fig. 4a that are not located directly in nadir view of the sensor are displaced away from the sensor. Instead of being mapped to *P'*, *P* is mapped to *PO* in the image. The degree of displacement depends on the distance between a building and the sensor's nadir point as well as on a building's height. The further away an elevated object *P* is located from the nadir axis of the optical sensor (increasing θ_2) and the higher it is, the more the building roof is displaced. The higher *P* is, the further away *P* is located from the optical nadir axis and the greater the off-nadir angle θ_1 becomes, the longer the distance between *PO* and *PS* will get.

The optical data was ortho-rectified by means of a DTM in order to reduce image distortions due to terrain undulations. Building façades stay visible and roofs are displaced away from the sensor nadir point since buildings are not included in the DTM. Such displacement effect can be seen in Fig. 4b to 4d. In Fig. 4b the building in the optical image is overlaid with its cadastral boundaries. The building roof is displaced to the right since the sensor nadir point is located on the left. The upper right part of the building is more shifted to the right than the lower left part because it is higher (see Fig. 4d for building height). Fig. 4c shows the same cut-out overlaid with the corner line extracted from the corresponding InSAR cut-out. Such corner line represents the location where the building wall meets the ground which can nicely be seen in Fig. 4d. Due to the previously outlined perspective effect the building roof falls to the right over the corner line. This effect is of high interest and can be exploited for three-dimensional modelling of the scene (Inglada and Giros, 2004, Wegner and Soergel, 2008) because the distance between the corner line and the building edge comprises height information.

4.2 Joint classification framework

A joint classification is carried out after having projected the optical and the InSAR primitive objects to the same ground geometry. In order to combine the building hints from optical and InSAR data, a fusion step is required. One possibility is data fusion in a Bayesian framework while another would be Dempster-Shafer evidential theory (Klein, 2004). Both approaches are usually requiring an object to be represented identically in the different sensor outputs, i.e., exactly the same region is found in both datasets but with slightly different classification results. This requirement is not met in the case of the combination of line features from InSAR data with roof regions from optical imagery.

Hence, combined analysis is based on the linear regression classifier already used for building extraction from optical data in (Mueller and Zaum, 2005). All potential building objects from the optical image are evaluated based on a set of optical features described in section 2.2 and on the InSAR corner line objects. The evaluation process is split up into two parts, an optical part and an InSAR part. Optical primitive objects are believed to contribute more information to building detection and hence their weight is set to two thirds. InSAR data is assumed to contribute less information to overall building recognition and thus the weight of primitive objects derived

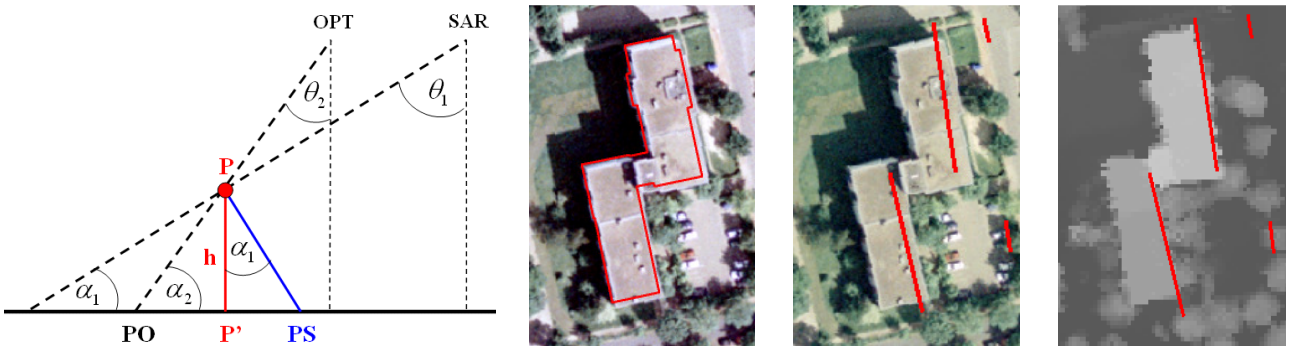


Figure 4. Comparison of SAR and optical viewing geometry under the assumption of locally flat terrain (a); optical data (b) overlaid with cadastral building footprint; optical data (e) and LIDAR data (d) overlaid with detected building corner

from SAR data is set to one third. Such weights are determined empirically and lead to good results. However, further research has to be done in order to support this choice with reasonable statistics.

A quality measure is assigned to each region and initially set to 1. In the first evaluation part each primitive is evaluated based on the optical feature vector. Each time a feature does not completely support the building hypothesis, the quality measure is reduced by multiplication with a value between 0 and 1. The exact reduction value for each feature was learned on manually labelled training data. Such reduced quality measure is again multiplied with another reduction value if another feature partly rejects a building hypothesis. The final quality measure based on the optical feature vector is weighted with 0.666.

A second region evaluation is conducted based on the corner line primitives extracted from InSAR data. First, all building object hypotheses are enlarged by two subsequent dilation operations. In this manner, a two-pixel wide buffer, corresponding to 0.6 meters in ground geometry, is added to the original region since building roofs may be shifted away from the corner line. Thereafter, it is checked if the corner line crosses this enlarged region with a certain minimum length. The initial quality measure is multiplied with a reduction value like in the optical case if this is not the case. The resulting quality measure based on the corner line is multiplied with a weighting factor of 0.333.

Finally, the overall quality measure is obtained by summing up the optical and the InSAR quality measures. In case neither an optical feature nor an InSAR feature has decreased the quality measure, both quality measures sum up to one. All regions that have a quality measure greater than an empirically determined threshold are classified as building objects. Such threshold was set to 0.6. As a consequence, a region may be classified as building region even if there is no hint from the InSAR data, but strong evidence from the photo. The reason is that some buildings do not show corner lines due to an unfavourable orientation towards the SAR-sensor (see the gabled-roofed buildings in the lower right corner of Fig. 5b) or occlusion of the potential corner line region by plants. On the contrary, a region cannot be evaluated as building region based merely on the corner line which are strong hints for buildings but may also caused by other abrupt height changes in urban areas.

5. RESULTS

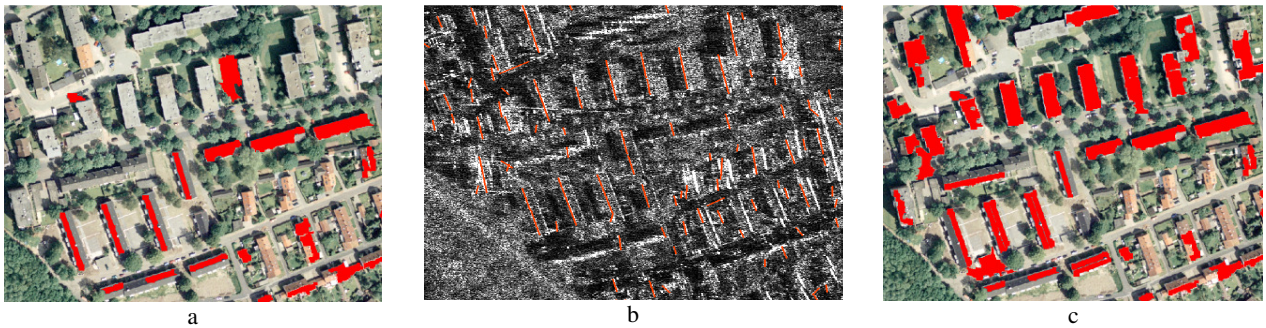
The InSAR data used in this project was recorded by the AeS-1 sensor of Intermap Technologies. The spatial resolution in range is about 38 cm while 16 cm resolution is achieved in azimuth direction. The two X-Band sensors were operated with an effective baseline of approximately 2.4 m. The mapped residential area in the city of Dorsten in Germany is characterized by a mixture of flat-roofed and gable-roofed buildings and low terrain undulation.

Results of the presented approach for building recognition by means of feature combination from optical imagery and InSAR data are shown in Fig. 5. In Fig. 5a building recognition results based solely on optical features are displayed. All parameters where specifically adjusted in order to achieve the lowest possible false alarm rate while still detecting buildings. Less than 50% of the buildings contained in the displayed scene are detected. In addition, false alarms could not be avoided completely. Results are rather poor due to the assumption that roofs do not split up into more than two regions during the region growing step, which is not met for the data at hand. As a consequence, several gable-roofed buildings with reddish roofs in the lower right corner of the image could not be recognized. Some big flat-roofed buildings in the upper part of the image are not detected because their colour and shape are similar to such of street segments. Thus, their evaluation value does not exceed the threshold.

Fig. 5b shows the corner lines extracted from the InSAR data superimposed onto one SAR magnitude image. An InSAR corner line could be detected for almost all buildings in this scene. Some lines are split into two parts because the corresponding building was partly occluded by, e.g., plants. Some corner lines in the lower right diagonally cross buildings which is not plausible. Most likely this effect is an artefact introduced by too large tolerances applied in the merging and prolongation steps of adjacent line segments. The final building recognition result using both optical and InSAR features is shown in Fig. 5c. The overall building recognition rate could be significantly improved to approximately 80% by integration of the InSAR corner lines into the classification procedure. Additionally, all false alarms could be suppressed. However, the gable-roofed buildings in the lower right corner stay undetected although InSAR corner lines are present. Such missed detections are due to the over-segmentation of the rather inhomogeneous roof regions in the optical image.

6. CONCLUSION AND OUTLOOK

In this work, first building detection results from combined optical and InSAR data on feature level were presented. A rather simple approach for feature fusion was introduced leading to a significantly improved building recognition rate. Additionally, the number of false alarms could be reduced considerably by the joint use of optical and InSAR features. Corner lines from InSAR data proved to be essential hints for



a b c

Figure 5. Results of building detection based on optical data (a), detected corner lines in the InSAR data (b), and of building detection based on InSAR and optical data (c)

buildings. Such corner lines also appear in single SAR images and hence this approach is not limited to InSAR data.

Further developed, this approach may be the basis for a change detection method after natural hazards like flooding and hurricanes. An optical image acquired before the hazard and SAR data acquired afterwards can be analyzed using the presented approach. A human interpreter would only have to check those buildings for damages that were not detected from both data sources. Hence, all buildings recognized from the combination of optical and SAR features, shown in red in Fig. 5c, would be classified as undamaged. Only buildings in the optical image that where not detected would have to be checked speeding up the entire damage assessment step significantly.

Although first results are encouraging, further improvements have to be made. One main disadvantage of the presented classification approach is that its quality measures are not interpretable as probabilities in a Bayesian sense. Although many parameters have been learned from training data, parts of the approach are still ad-hoc. A next step will thus be the integration of the presented approach into a Bayesian framework.

Furthermore, the differences of the sensor geometries should be used for further building recognition enhancement. Since the roofs of high buildings are displaced away from the sensor and parts of the façade appear in the image, roof regions have to be shifted towards the sensor in order to delineate building footprints. Such displacement also bears height information which may be used as an additional feature for building recognition. More height information may also be derived directly from the InSAR data.

Finally, three-dimensional modelling of the scene could be accomplished based on the building footprints, a height hypothesis and maybe even the estimation of the roof type. An iterative joint classification and three-dimensional modelling in a Bayesian framework, including context information, will be the final goal of this project.

7. REFERENCES

Inglada, J., and Giros, A. 2004. On the possibility of Automatic Multisensor Image Registration. IEEE Transactions on Geoscience and Remote Sensing, Vol. 42, No. 10, pp. 2104-2120.

Klein, L. A. 2004. Sensor and Data Fusion-A Tool for Information Assessment and Decision Making, 3rd ed., Bellingham, WA: SPIE Press, pp.127-181.

Mueller, S., and Zaum, D. W. 2005. Robust Building Detection in Aerial Images. IntArchPhRS, Vol. XXXVI, Part B2/W24, pp. 143-148.

Soergel, U., Thiele, A., Cadario, E., Thoennessen, U. 2007. Fusion of High-Resolution InSAR Data and optical Imagery in Scenes with Bridges over water for 3D Visualization and Interpretation. In: *Proceedings of Urban Remote Sensing Joint Event 2007 (URBAN2007)*, 6 pages.

Thiele, A., Schulz, K., Thoennessen, U., Cadario, E. 2006. Orthorectification as Preliminary Step for the Fusion of Data from Active and Passive Sensor Systems. In: *Proceedings of the IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems 2006 (MFI2006)*, 6 pages.

Thiele, A., Cadario, E., Schulz, K., Thoennessen, U., Soergel, U. 2007. Building Recognition From Multi-Aspect High-resolution InSAR Data in Urban Areas. IEEE Transactions on Geoscience and Remote Sensing, Vol. 45, No. 11, pp. 3583-3593.

Thiele, A., Cadario, E., Schulz, K., Thoennessen, U., Soergel, U. 2008. Reconstruction of residential buildings from multi-aspect InSAR data. In: *Proceedings of ESA-EUSC Workshop*, Frascati, Italy, available http://earth.esa.int/rtd/Events/ESA-EUSC_2008/, 6p.

Tupin, F., Maitre, H., Mangin, J.-F., Nicolas, J-M., Pechersky, E. 1998. Detection of Linear Features in SAR Images: Application to Road Network Extraction. IEEE Transactions on Geoscience and Remote Sensing, Vol. 36, No. 2, pp. 434-453.

Tupin, F., and Roux, M. 2003. Detection of building outlines based on the fusion of SAR and optical features. ISPRS Journal of Photogrammetry and Remote Sensing, Vol. 58, pp. 71-82.

Tupin, F., and Roux, M. 2005. Markov Random Field on Region Adjacency Graph for the Fusion of SAR and Opitcal Data in Radargrammetric Applications. IEEE Transactions on Geoscience and Remote Sensing, Vol. 42, No. 8, pp. 1920-1928.

Wegner J.D., and Soergel, U. 2008. Bridge height estimation from combined high-resolution optical and SAR imagery. IntArchPhRS, Vol. XXXVII, Part B7-3, pp. 1071-1076.

Xu, F., Jin, Y.-Q. 2007. Automatic Reconstruction of Building Objects From Multiaspect Meter-Resolution SAR Images. IEEE Transactions on Geoscience and Remote Sensing, Vol. 45, No. 7, pp. 2336-2353.

FAST VEHICLE DETECTION AND TRACKING IN AERIAL IMAGE BURSTS

Karsten Kozempel and Ralf Reulke

German Aerospace Center (DLR e.V.), Institute for Transportation Systems
Rutherfordstraße 2
12489 Berlin
karsten.kozempel@dlr.de, ralf.reulke@dlr.de

KEY WORDS: aerial, image, detection, tracking, matching

ABSTRACT:

Caused by the rising interest in traffic surveillance for simulations and decision management many publications concentrate on automatic vehicle detection or tracking. Quantities and velocities of different car classes form the data basis for almost every traffic model. Especially during mass events or disasters a wide-area traffic monitoring on demand is needed which can only be provided by airborne systems. This means a massive amount of image information to be handled. In this paper we present a combination of vehicle detection and tracking which is adapted to the special restrictions given on image size and flow but nevertheless yields reliable information about the traffic situation.
Combining a set of modified edge filters it is possible to detect cars of different sizes and orientations with minimum computing effort, if some a priori information about the street network is used. The found vehicles are tracked between two consecutive images by an algorithm using Singular Value Decomposition. Concerning their distance and correlation the features are assigned pairwise with respect to their global positioning among each other. Choosing only the best correlating assignments it is possible to compute reliable values for the average velocities.

1 INTRODUCTION

1.1 Motivation

The gathering of traffic information is a base for all kinds of traffic modeling, simulation and prediction for tasks like emission reduction, efficient use of infrastructure or extension planing of the road network as well as the intervention and resource planing. Next to the use of inductive loops, Video Image Detection Systems (VIDS) have become a common alternative due to their low price as well as their simplicity and effort of installation. Furthermore inductive loops can't cover the whole road network and a lot of data has to be estimated. Especially during mass events or disasters with huge congestions or road blocks, they can't yield reliable information.
For this special purpose the German Aerospace Center (DLR e.V.) developed the ANTAR system for airborne traffic monitoring on demand. During the soccer world cup 2006 it was successfully applied to gather traffic data and predict traffic situation in three German cities (Ruhé et al., 2007). Based on this the DLR is developing the ARGOS system for wide-area traffic monitoring (fig.1). It contains next to a radar system the 3K-Cam, a device of three digital cameras with 16 mega pixels each. Together they cover an area of 2,5 km x 0,7 km with a resolution of 20 cm at an altitude of 1000 m over ground. Additionally a GPS/IMU-unit is used to record positioning and orientation data for every image taken. Thereby the achieved image data gets orthorectified and georeferenced on-board which means that the images arriving the traffic detecting software can be used as map images with given orientation and scale. A fact that makes measuring distances and computing velocities less complex.
In the first chapter the conditions related to the observation system are explained as well as the published work on this area. The second chapter describes the used algorithms, a modified edge filter for fast vehicle detection and an extended singular value decomposition concerning distances and correlations for tracking in very short sequences. After this the results with a few examples are presented. Finally a conclusion with considering possible further research will close the paper.

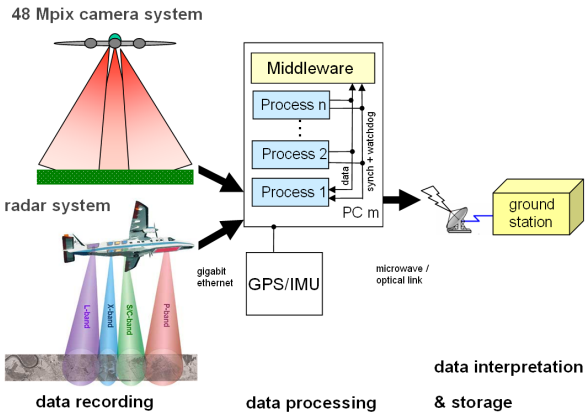


Figure 1: Traffic monitoring system ARGOS

1.2 Special conditions

There are two special points to consider while developing detection and tracking. It should be respected that the preprocessed images depending on their altitude over ground can be very large, in the shown case 25-30 mega pixels. That's why the detecting algorithm should be rather fast than exact. Already the previous system ANTAR demonstrated that for an overview of the traffic situation a completeness of two thirds is acceptable.
Due to the mentioned size of the images (original size is 16 mega pixels) they cannot be transmitted continuously. After a burst of a few images (2-4) the stream is cut to save them. Therefore it is not necessary to implement a complex tracking filter which needs a long period to adapt to the scene.

1.3 Related work

A grand variety of approaches in vehicle detection as well as in object tracking has been released in the last years.
Detection methods can be divided into two groups, depending on the kind of model being used. The use of explicit models

for example is explained in (Haag and Nagel, 1999), (Moon et al., 2002), (Hinz, 2004) and (Ernst et al., 2005). In (Haag and Nagel, 1999) a very extensive database of about 400 different three-dimensional car models is used to predict the appearance of vehicles including their shadow cast. In (Hinz, 2004) the author uses not only the shadow but additionally the luminance and reflectivity of the car's surface as well which of course is more expensive to process. Next to shape and shadow in (Zhao and Nevatia, 2003) they try to recognize the windshield of vehicles. The final decision is made by a Bayesian Network. Most of them have a very reliable detection rate of more than 90 percent but a long computing time. In the papers (Moon et al., 2002) and (Ernst et al., 2005) they use rather simple two-dimensional models for detection. While in (Ernst et al., 2005) the authors search in the edge filtered image for rectangular objects of certain size in (Moon et al., 2002) they already shape the edge filter to a rectangle of expected car size. Both of them provide a fast and acceptable detection rate using additional information about street area and direction.

The use of implicit models is explained in (Grabner et al., 2008) and (Lei et al., 2008). In (Grabner et al., 2008) the author supposes to use a learning AdaBoost algorithm which is robust and fast by making a lot of cascaded weak decisions. In (Lei et al., 2008) they train a support vector machine with the SIFT descriptors of selected cars and non-cars. But both of these approaches have to be trained with lots of positive and negative samples before working independently. Additionally it is not easy to cover all cases of illumination and environment. That's why many leaning algorithms have to be trained for every situation separately. Another easy approach for detection of moving cars without using any model is explained in (Reinartz et al., 2006) where they detect all moving objects in adjacent images by computing the normalized difference image. But as the georegistration of the images often is less exact than the pixel size, the images have to be coregistered first. On the other hand only moving objects can be detected while traffic jams or queues in front of a traffic light would be ignored.

Concerning tracking there are lots of publications using optical flow and Kalman or particle filters to predict the expected displacement and appearance in following images. (Haag and Nagel, 1999) and (Nejadasl et al., 2006) pursued this approach which is not easy to realize in the special case of only two or three adjacent images. In (Lenhart and Hinz, 2006) they use especially triplets of images to determine the best match between at least three states which can be described as a kind of prediction. Another good idea for the special case of very short bursts is presented in (Scott and Longuet-Higgins, 1991) and improved in (Pilu, 1997). The authors use singular value decomposition of a distance matrix to match a group of features to another one with respect to the relative positions of all features among each other. (Pilu, 1997) later extends the approach by adding the correlation between pairs of features.

2 APPROACH

2.1 Preprocessing

To identify the active regions as well as the orientation of images among each other they have to be georeferenced, which means their absolute geographic position and dimension have to be defined. Related to the GPS/IMU information and a digital terrain model the image data gets projected into GeoTIFF images, which are plane and oriented into north direction. This is useful to combine the recorded images with existing datasets like maps or street data. To avoid examining the whole image data, only the street area given by a database is considered.

2.2 Detection

For providing fast detection of traffic objects in the large images a set of modified edge filters, that represent a two-dimensional car model, is used. Recent tests showed that the car's color information does not yield better results in detection than its gray value. Therefore the original images are converted into gray images. This conversion saves two thirds of filtering time. As there is additional information about street area and orientation this knowledge is used as well. The databases provided by Navteq (www.navteq.com) and Atkis (www.atkis.de) for example contain that information about the street network. For every street segment covered by the image a bounding box around it is cut out. The subimage is masked with the street segment to only use the filters on traffic area. We use neither a Hough transformation for finding straight edges nor a filter in shape of the whole car, as mentioned in (Moon et al., 2002). But we create four special shaped edge filters to represent all edges of the car model, which are elongated to the average expected size and turned into the direction given by the street database (fig.2 and 3). To

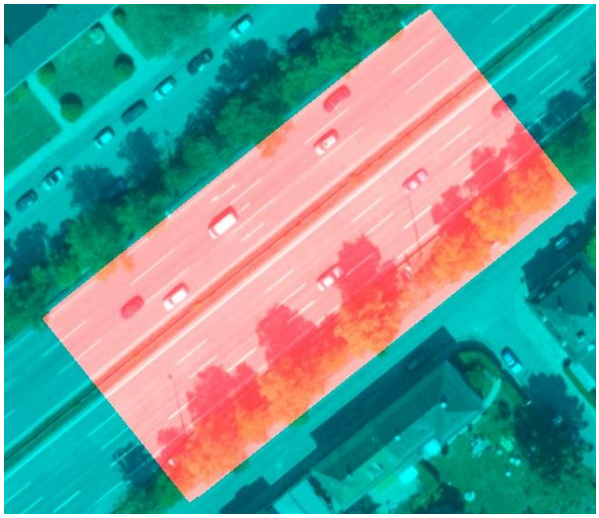


Figure 2: Mask based on Navteq street segments

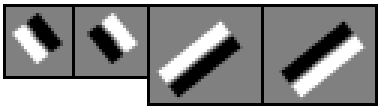


Figure 3: The associated filter kernels

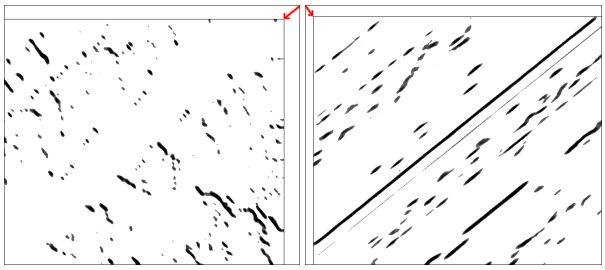


Figure 4: The shifted and thresholded filter answers 2 and 3

avoid filtering for all different car sizes, we only shift the filter answers (fig.4) to the expected car edges within a certain range. This has the same effect as positioning the filter kernels around an anchor point. In the conjunction image of the four thresholded and shifted edge images remain blobs at the position, where all four filters have answered strong enough to the related edge filter. The regions remaining (fig.5) become thinned by a non-maxima

suppression until one pixel each is left representing the car's center. Fig.6 shows the regions left related to the cars that caused them.
For bigger vehicles like trucks the same filter answers are used. To recognize long edges without using new filters, the given answers of the side edges are shifted along the side of the car and always conjuncted with each other.
To avoid cars being detected twice, all observations are tested pairwise for their distances among each other. Some observations have more than one maximum, or vehicles are detected twice between two neighboring street segments. With respect to their size and orientation, objects below a certain distance to each other are discarded while only the one with the strongest intensity remains.

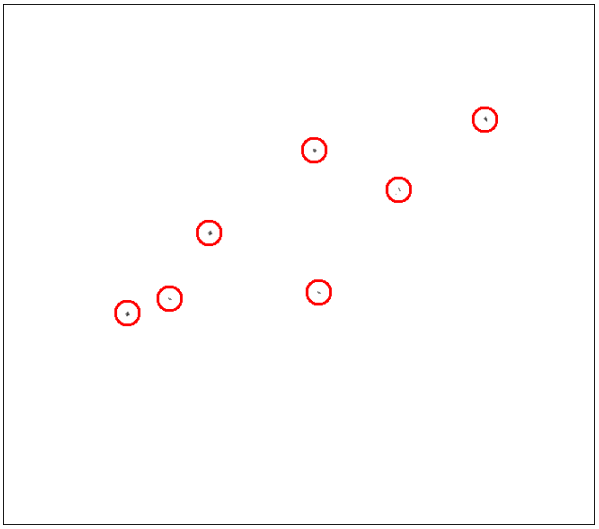


Figure 5: Regions where all filters answered

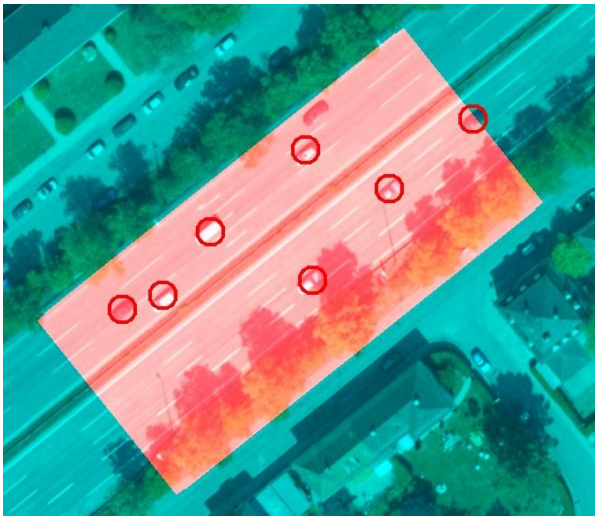


Figure 6: The detected cars

2.3 Tracking

As there are only short bursts of images, a classic Kalman filter cannot really be used. As already mentioned Lenhart's approach in (Lenhart and Hinz, 2006) uses prediction for image triplets. This works just in case there are triplets. Bursts with less than three images, which appear as well, have to be handled different. That's why we only consider relations between two consecutive images. Scott and Longuet-Higgins suggest in (Scott and Longuet-Higgins, 1991) a singular value decomposition as a kind of one-to-one correspondence with respect to the positions of all neighboring objects. This is more an association than a real tracking as only the last image's information is used. If I and J are two images with m features I_i and n features J_j we build a proximity matrix $\mathbf{G}$ with the Gaussian-weighted distances G_{ij} between every feature I_i and J_j.

$$G_{ij} = e^{-r_{ij}^2/2\sigma^2} \tag{1}$$

where $r_{ij} = ||I_i - J_j||$ is is the euclidean distance. So the elements G_{ij} decrease monotonically with the distance. The parameter σ defines the degree of interaction between the features. A small value enforces local and a big one rather global interaction. It is recommended to choose σ as large as the average expected distance the feature pairs have.
The next step is to perform a singular value decomposition of the proximity matrix $\mathbf{G}$. The Algorithm is provided by a lot of software libraries. Here the one in OpenCV was used.

$$\mathbf{G} = \mathbf{TDU}^T \tag{2}$$

After the SVD the matrices $\mathbf{T}$ and $\mathbf{U}$ are orthonormal matrices and the diagonal matrix $\mathbf{D}_{m\times n}$ contains the positive singular values as diagonal elements in descending order. As the third and last step a new matrix $\mathbf{P}$ has to be computed by

$$\mathbf{P} = \mathbf{TEU}^T \tag{3}$$

where $\mathbf{E}$ is the changed diagonal matrix $\mathbf{D}$ with all elements replaced by $\mathbf{1}$. The resulting matrix $\mathbf{P}$ has the same dimensions as $\mathbf{D}$ but by the algorithm the values P_{ij} for good pairings have been amplified while those for bad ones have been reduced. So if P_{ij} is the greatest element in column and row the two features I_i and J_j are in a 1:1 correspondence with one another.
Furthermore Pilu (Pilu, 1997) extends the algorithm for feature-based stereo matching by using the cross correlation of two features next to their distance. So the SVD-association can be used for images concerning the similarity of a certain window around their features. Adding this (Gaussian-weighted) information to the proximity matrix $\mathbf{G}$ the elements G_{ij} result as follows:

$$G_{ij} = e^{-(C_{ij}-1)^2/2\gamma^2} \cdot e^{-r_{ij}^2/2\sigma^2} \tag{4}$$

where the left term is the Gaussian-weighted function of the normalized correlation coefficient C_{ij} between the features I_i and J_j. The parameter γ determines how fast the values decrease with C_{ij}. During our tests the best values lie between 0.4 and 1.0.

3 RESULTS AND DISCUSSION

3.1 Detection

The computing time and the accuracy of detection always depend on the number, size and quality of street segments given by the database. In the first example (shown in fig.6) only a broad highway in Munich has been tested without any smaller streets being considered. The processing of the 28 mega pixels large image took 30 seconds (Athlon 64 X2, 2.2 GHz, 2 GB RAM). The 96 vehicles were counted manually as ground truth and compared with the detected vehicles. The varying detection rates caused by varying thresholds are shown as the red graph in fig.7 and 8. As one can see there is always a trade-off between completeness and correctness. The more sensitive the thresholds are set the more false positives they will find. The graph shows the detection rate (number of true detected cars/real number of cars) in

relation to the rate of false positives (number of false detected objects/number of detected objects). To be honest the false positives rate is not very objective, as the number of false detected objects does not depend on the real number of cars, and could turn out very bad just in case there is only one car in the image. Therefore in (Lei et al., 2008) they consider the FP-number in relation to the length of streets. A still better way would be to take the street area, for example 'false positives per hectare'. In this example there is an optimal point, where detection reaches 80 percent while the FP-rate is only ten percent or one car per hectare.

A rather bad sample (the worst in our evaluation) represents the

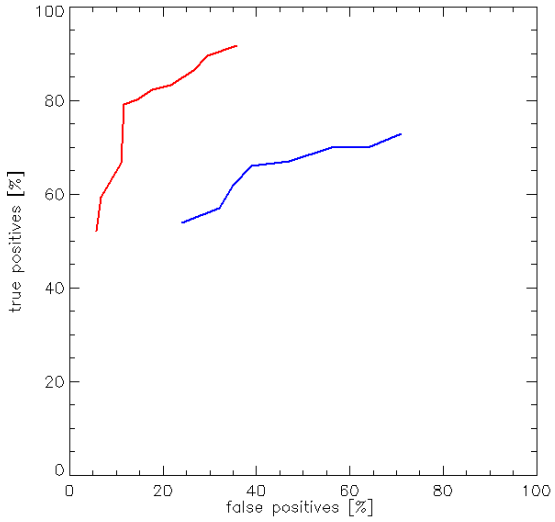


Figure 7: Detection rates on a highway (red) and narrow streets (blue) depending on false positives per detected cars

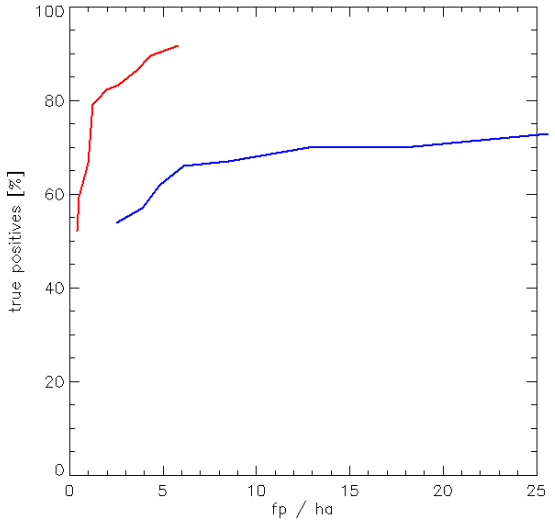


Figure 8: Detection rates on a highway (red) and narrow streets (blue) depending on false positives per area

blue graph in fig.7 and 8 where more than 300 cars have been clicked by hand. If we consider streets of all sizes in the Munich suburban area, on the one hand the detection time takes longer (more than 60 seconds) and the results become worse as well. The detection rate stays around two thirds while only the number of false positives rises from 5 up to 25 per hectare.

A reason for the bad detection rate in the second example is the accuracy of street coordinates. As many smaller street elements are drawn next to the real street (fig.9) the algorithm misses many cars while detecting some rectangular structures next to the street. An approach to avoid this might be to improve the street accuracy by alternative street databases or street detection which should not be considered in this paper.

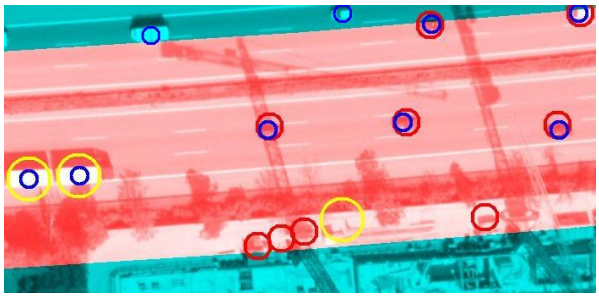


Figure 9: False positives and negatives due to incorrect coordinates (blue - existing car, red - found car, yellow - found truck)

3.2 Tracking

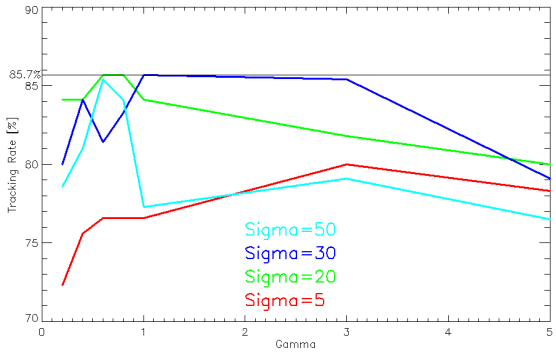


Figure 10: Correctness rate of tracks depending on the parameters σ and γ

We implemented the tracking algorithm as explained above by using the vehicles distances on UTM-projection and the normed correlation coefficient of all three color channels in a 20-by-20-pixels window around them. As the images cover an area of 700 on 1000 meters with hundreds of cars each, it is not easy to show how the whole set of tracks looks. That's why only one street was picked out for visualization. In fig.10 the resulting tracking rates depending on the parameters σ and γ are shown. As one can see the best results we get if σ is between 20 and 30. If the value is too small ($\sigma = 5$) the dependence of the positions among each other is not respected enough. This results not only in incorrect assigned pairs but also in crude mistakes by assigning objects together which are located very far from each other. This can strongly falsify the measured velocities. Furthermore γ should neither be too small nor too high. The best results yield values between 0.4 and 1.0. Around these settings a correctness of more than 80 percent (best value 85.7%) is achieved.

As for the average velocities it is rather important to accept correct tracks than getting all vehicles tracked, after the SVD the acceptance is bound to the correlation coefficient of a pairing. If the pairing next to its ranking in row and column does not pass a threshold for the CC, it is discarded although it might be correct. In fig.11 the remaining tracks are shown. In the upper half of the image 49 objects have been detected. 39 of them have been detected in the lower half as well which means they are possible to track. 36 of the objects have been assigned to another one, 30 of them were assigned correctly. After the thresholding with a CC of 0.9 still 26 of the 36 tracks remain. So from end to end

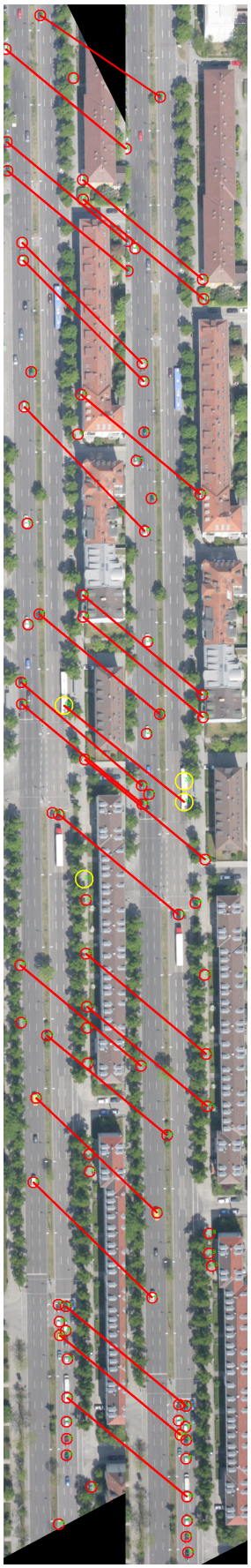


Figure 11: Tracked objects filtered by CC-threshold 0.9 (100% correct)

only 53 percent (26 out of 49) of all detected objects are found again and tracked, but with a correctness of 100 percent. Surely there should be more tests with more representative numbers, but we did not have enough reliable reference data yet. This will be done in the near future.

4 CONCLUSIONS AND FUTURE WORK

In this paper we presented a car detecting and a tracking algorithm which have been especially chosen and adapted to the given situation, the flying traffic monitor ARGOS. It was shown how they work and that they brought satisfying results depending on the environmental conditions. Furthermore it was shown, where the approach has problems and continuous work can be done.
Surely the system can be improved in some points and a few of them should be given here. First of all the street accuracy problem which could be easily solved by using another database. And it should be mentioned that there was already the attempt to use the more accurate street database Atkis. On the one hand the coordinates were indeed more exact and yielded slightly better detection rates, but on the other hand the database divides the street network into too small segments, which take a lot more time to process one by one. Additionally the achieved data should be mapped on Navteq segments, which would not be easy. So the next step is to integrate the newest version of the Navteq database being bought at the time.
Furthermore the edge detection could be optimized for example by running it on the GPU, but it has not been considered so far. Another idea is to compute the filtering in the frequency space. The Fourier-transformed images and filters just have to be multiplied in frequency space and transformed back. The only problem is that the filters change with every street segment, so there are four filters and four filtered images to be transformed every time. The approach was already explored, but the Fourier-transformation implemented in OpenCV needs longer than direct convolution, because it uses floating point numbers.
Next to this the detected cars could be verified by a more expensive algorithm like a Bayesian Network or a Support Vector Machine because some of the false positives do not look like a car at all. So they would be easy to discard.

ACKNOWLEDGEMENTS

The authors would like to thank Dr. Franz Kurz and Dr. Dominik Rosenbaum (Remote Sensing Technology Institute, German Aerospace Center, Oberpfaffenhofen) for providing the GeoTIFF images and navigation data.

REFERENCES

Ernst, I., Hetscher, M., Thiessenhusen, K., Ruhé, M., Börner, A. and Zuev, S., 2005. New approaches for real time traffic acquisition with airbone systems. Int. Archives of Photogrammetry, Remote Science and Spatial Information Sciences 36, pp. 68–73.

Grabner, H., Nguyen, T. T., Gruber, B. and Bischof, H., 2008. On-line boosting-based car detection fron aerial images. ISPRS Journal of Photogrammetry and Remote Sensing.

Haag, M. and Nagel, H., 1999. Combination of edge element and optical flow estimates for 3d-model-based vehicle tracking in traffic image sequences. International Journal of Computer Vision 35, pp. 295–319.

Hinz, S., 2004. Detection of vehicles and vehicle queues in high resolution aerial images. Photogrammetrie - Fernerkundung - Geoinformation (PFG) 3/04, pp. 201 – 213.

Lei, Z., Li, D. and Fang, T., 2008. Vehicle detection in high-resolution satellite imagery using sift features and support vector machines. ISPRS Journal of Photogrammetry and Remote Sensing.

Lenhart, D. and Hinz, S., 2006. Automatic vehicle tracking in low frame rate aerial image sequences.

Moon, H., Chellappa, R. and Rosenfeld, A., 2002. Performance analysis of a simple vehicle detection algorithm. Image and Vision Computing 20, pp. 1–13.

Nejadasl, F. K., Gorte, B. G. H. and Hoogendoorn, S. P., 2006. Optical flow based vehicle tracking strengthened by statistical decisions. ISPRS Journal of Photogrammetry and Remote Sensing 61, pp. 149–158.

Pilu, M., 1997. Uncalibrated stereo correspondence by singular value decomposition. Technical report, Digital Media Department, HP Laboratories Bristol.

Reinartz, P., Lachaise, M., Schmeer, E., Krauss, T. and Runge, H., 2006. Traffic monitoring with serial images from airborne cameras. ISPRS Journal of Photogrammetry and Remote Sensing 61, pp. 149–158.

Ruhé, M., Kühne, R., Ernst, I., Zuev, S. and Hipp, E., 2007. Airborne systems and data fusion for traffic surveillance and forecast for the soccer world cup. Technical report, German Aerospace Center.

Scott, G. L. and Longuet-Higgins, H. C., 1991. An algorithm for associating the features of two images. Proc. Royal Society London 244, pp. 21–26.

Zhao, T. and Nevatia, R., 2003. Car detection in low resolution aerial images. Image and Vision Computing 21, pp. 693–703.

REFINING CORRECTNESS OF VEHICLE DETECTION AND TRACKING IN AERIAL IMAGE SEQUENCES BY MEANS OF VELOCITY AND TRAJECTORY EVALUATION

D. Lenhart[1], S. Hinz[2]

[1]Remote Sensing Technology, Technische Universitaet Muenchen, 80290 Muenchen, Germany
Dominik.Lenhart@bv.tu-muenchen.de
[2]Institute of Photogrammetry and Remote Sensing, University of Karlsruhe, 76128 Karlsruhe, Germany
Stefan.Hinz@ipf.uni-karlsruhe.de

KEY WORDS: Traffic Monitoring, Vehicle Trajectories, Aerial Image Sequences, Fuzzy Logic, Evaluation

ABSTRACT:

Derivation of statistical traffic data is highly dependent on the balance of detection and false alarm rates. In case false alarms have not been eliminated in the initial detection phase, they are often subsequently tracked, though, resulting in trajectories that do not match the true traffic situation. This finally leads to derivation of erroneous traffic parameters within the individual road segments. In this paper, a method is described how to eliminate false alarms by evaluating the trajectories and velocities of a tracking procedure. Basically, two types of false alarms are considered which bias the statistics of traffic data: The first type deals with redundant detections that lead to multiple trajectories biasing the statistics. The second type comprises false alarms that belong to the static background inducing zero-velocity into the statistics. We show that the presented procedure is able to increase the total correctness of detection and tracking from 65% up to 95% which allows a much more precise calculation of traffic flow parameters.

1. TRAFFIC MONITORING

The task of collecting wide area traffic parameters plays important role in today's traffic management. Aerial images offer a complement source to common measurement systems like induction loops and stationary video cameras. Besides giving a visual overview, image sequences which cover large areas can deliver a time snapshot of a spatially fully covered traffic situation of the recorded region.

In recent years, traffic monitoring using air- and space images became more and more attractive mainly due to the availability of cost-effective and flexible high-resolution systems mounted on aircrafts, i.e. the LUMOS/ANTAR system for traffic monitoring (Ernst et al., 2003; Ernst et al. 2005; Ruhé et al., 2007) or the 3K camera system (Kurz et al., 2007), or on HALE platforms and UAVs as presented in the Pegasus project (Everaerts et al., 2004). An extensive overview on recent developments is given, for instance, in (Stilla et al., 2005; Hinz et al., 2006; Lenhart et al., 2008). The following methods are especially designed for traffic monitoring with DLR's 3K camera system. This system is able to capture image sequences with a frame rate of approx. 3Hz – 7Hz depending on the imaging mode (continuous imaging or bursts) with a spatial resolution of 20cm – 50cm depending on the flight height. Concepts for deriving traffic data from these aerial image sequences have been proposed in (Rosenbaum et al. 2008) and (Lenhart et al. 2008). The traffic parameters which are calculated from image sequences are namely the mean velocity and traffic density per road segment. The resulting parameters are then integrated into traffic flow models such as the DELPHI traffic portal illustrated in (Behrisch et al.).

2. INFLUENCE OF FALSE ALARMS

Detection methods as proposed in (Rosenbaum et al. 2008) or (Lenhart et al. 2008) deliver a detection quality of about 60% completeness and 65-75% correctness. False alarms are mainly caused by structures which appear similar to vehicles, like i.e. belonging to shadows, road banks etc.

The influence of the false alarm rate on the calculation of generic traffic parameters can be studied using, e.g., Monte-Carlo simulations. In the following experiment a dense traffic scenario on a multi-lane highway was captured with an image sequence and all car trajectories were manually measured in this sequence, eventually leading to mean velocity profiles for each lane of the highway. Then, a predefined percentage of detections were selected at random positions along the road and contaminated with a specific percentage of random false alarms. Based on these data the velocity profiles were calculated for each lane again and compared to the reference data. As the estimation of the velocity profile depends strongly on the randomly selected positions of the cars, these experiments have been carried out 10000 times, in order to gain a certain statistic about the quality of the estimated profiles. The following table summarizes the RMS values and standard deviations for the estimated velocity profiles depending on the respective detection and false alarm rate.

50% detection rate 5% false alarm rate		50% detection rate 10% false alarm rate		50% detection rate 25% false alarm rate	
RMS [km/h]	σ [km/h]	RMS [km/h]	σ [km/h]	RMS [km/h]	σ [km/h]
5.22	**2.61**	**7.03**	**4.01**	**10.25**	**6.27**
30% detection rate 5% false alarm rate		30% detection rate 10% false alarm rate		30% detection rate 25% false alarm rate	
RMS [km/h]	σ [km/h]	RMS [km/h]	σ [km/h]	RMS [km/h]	σ [km/h]
5.97	**3.17**	**8.03**	**4.66**	**11.30**	**6.58**

Table 1: Monte-Carlo simulation of reconstruction of velocity profiles depending on detection and false alarm rates

As can be seen, especially the false alarm rate highly influences the quality of the estimates. For instance, it is still possible to reconstruct the velocity profile up to 6km/h ± 3km/h at a detection rate of only 30% when keeping the false alarm rate at 5%.

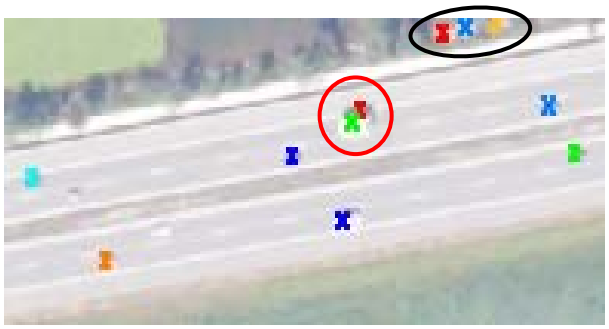


Figure 1: Detection result with false alarms. The red circle indicates redundant objects, the black mark shows objects belonging to the background.

There are mainly two ways of how false alarms are being tracked (see example in Figure 1):

- Collinear motion for redundant objects/features belonging to vehicles (trailer, car shadow or other).
- With zero velocity if objects belong to the background (road bank, shadows of trees etc.)

It is easy to see that these false alarm objects influence statistical traffic data in a manner that may lead to wrong conclusions of the traffic situation or to conflicts in model calculation.

To demonstrate such influence, two examples shall be mentioned:

- In a traffic scenario of a congestion where one lane moves slightly faster than the other (see Figure 2), false alarms belonging to (mainly larger) vehicles of the faster lane concurrently increase the derived density and raise the calculated average velocity. This obviously leads to a conflict in the traffic evaluation. If the false alarms belong to vehicles of the slower lane, the average velocity is lowered and thus implying an even higher vehicle density than there actually is.

- Let us assume a snapshot of a real situation of free flowing traffic with 30 cars moving with an average velocity of about 60 km/h (which corresponds to the speed limit). By assuming 60% completeness and 70% correctness, around 18 cars will be correctly detected and there will be 8 false alarms. In case that the false alarms belong to static background they will obtain a speed 0 km/h. This leads to a calculated average velocity of 41 km/h which implies rather dense traffic and thus feeding the traffic flow model with erroneous input data.

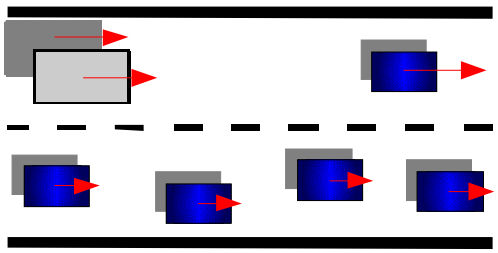


Figure 2: Congested traffic situation with different velocity in each lane.

Therefore, it is desirable to eliminate the false alarms of the initial detection to achieve a better quality of the calculated average velocity.

3. CONCEPT OF REFINEMENT

To improve the initial detection quality, we include generic knowledge about the velocity statistics and geometric layout of traffic flow in typical traffic situations (e.g. “free flowing”, “congestion”, “traffic jam”). To this end, we first track all initial detections and then eliminate the included false alarms based on an analysis of geometric layout and velocity of the trajectories.

3.1 Summary of tracking procedure

Initial vehicle candidates are extracted in the neighborhood of predefined road axes using a blob detection algorithm tuned for color images. Image triplets are then used for tracking, in order to gain a certain redundancy allowing an internal evaluation of the results. A vehicle image model is created by selecting a rectangle around a particular detection. By using the shape-based matching algorithm (Steger, 2001), car hypotheses are found in the successive images. The matching procedure delivers matches in Image 2 and in Image 3. Then, new car image models are created at all hypotheses positions in Image 2 and matched to Image 3. Of course, these matches may contain multiple match results. Finally, all results obtained in Image 3 are checked for consistency including a smoothness criterion of the trajectory to determine the correct combination of the matches. A detailed explanation of this approach can be found in (Lenhart et al., 2008).

The described tracking method is a very robust one delivering correct matches at about 99%, yet it tracks objects of any kind as long as their motion fulfills smoothness constraints similar to those of cars. Thus, trajectories of false initial detections are potentially tracked and also considered as “correct”. Based on the results of the tracking, the refinement is carried out.

3.2 Elimination of redundant objects

A first step to eliminate false detections is to remove redundant objects from the set of detections. These are the kind of objects that belong to vehicles, such as shadows or trailers.
For each pair of detections, the spatial distance is calculated. A search for very small distances delivers candidates for redundant objects. Since candidates may also include vehicles within a passing maneuver, these candidates need to be analyzed for their trajectories. The analysis includes the speed and direction of the determined trajectories and relative direction between the candidates. Identical trajectories and constant relative direction indicates redundant candidates while passing vehicles will have at least a slight difference in their speed or relative orientation.

It is now tested which of the redundant candidates is the car and which is the object to be eliminated. Therefore, a quick test of the gray or color value in the center of the objects is carried out. The darker and less colored object is assumed to be the shadow and is therefore eliminated from the set of detections. In case that both objects have a similar gray or color value, the trailing object is eliminated.

3.3 Knowledge representation of traffic situations

In order to evaluate the velocities of the vehicles we need to formulate our knowledge and expectations about the typical traffic situations such as "free flowing traffic", "congestions" or "traffic jams". In dependence of the state of traffic and the location with respect to intersections or traffic lights, different interactions between vehicles occur.
A well substantiated statistical concept like Bayes' theorem would provide a sound basis for evaluation. However, determining the probability density functions is hardly feasible because extensive and sufficient samples are missing. Hence, it is advisable to avoid a concept that claims statistical integrity.
In contrast to Bayes' theorem, fuzzy logic offers an intuitive method to represent knowledge of classes by easy parameterization (Zadeh, 1965). It is also frequently applied for modeling car following behavior (Brackstone and McDonald, 1999). Therefore, we decided to use fuzzy logic to describe our knowledge about traffic.

3.3.1 3D fuzzy membership function for active vehicles

Let us define a fuzzy set A that describes vehicles which are actively involved in traffic. Besides normally moving cars, these may be standing vehicles in traffic jams or waiting at red traffic lights or other crossroads.
Since we are only interested in the possibility of an object belonging to A, we neglect the alternative set $\bar{A}$ of inactive objects which may be false alarms of the detection, parking vehicles or erroneous tracks.

For the fuzzy set A, a membership function needs to be defined, indicating the possibility μ_A that a car belongs to A in dependence of its velocity v. However, μ_A also strongly depends on the traffic density D and the distance d from intersections. It is quite obvious, that in a free flowing situation in the middle of road segment the possibility that a car stands still is 0. In contrast to that, zero speed has a rather high possibility near intersections or in jam situations. In order to meet these different traffic situations, we have to consider the conditional possibilities $\mu_A(v|D,d)$. In the sequel, the units for the measures given shall be v [km/h], D [cars/km per lane] and d [m].

First, we should outline the ranges of D and d where μ_A may change significantly. A density of lower or equal to D = 30 corresponds to free flowing traffic while a density of D = 180 represents the maximum density of a traffic jam when there is almost no motion at all (Hall, 1999). Below 30, $\mu_A(v,D|d)$ remains constant.
The interesting range for d is approximately between 150 meters before an intersection because this describes the range where drivers start to brake and 50 meters behind the intersection where drivers accelerate until they reach their desired travel speed. Outside of the range of [-50m;150m], $\mu_A(v,d|D)$ is constant for all values of d.

Over the entire space of v, D and d, this results in a 3D membership function $\mu_A(v,d,D)$. Please note that the values also depend on the road type, speed limits intersection layout. For different road conditions, different functions have to be developed. The mentioned example function refers to a major city road with multiple lanes with a speed limit of 60 km/h and an intersection with a road of equivalent type controlled by traffic lights.

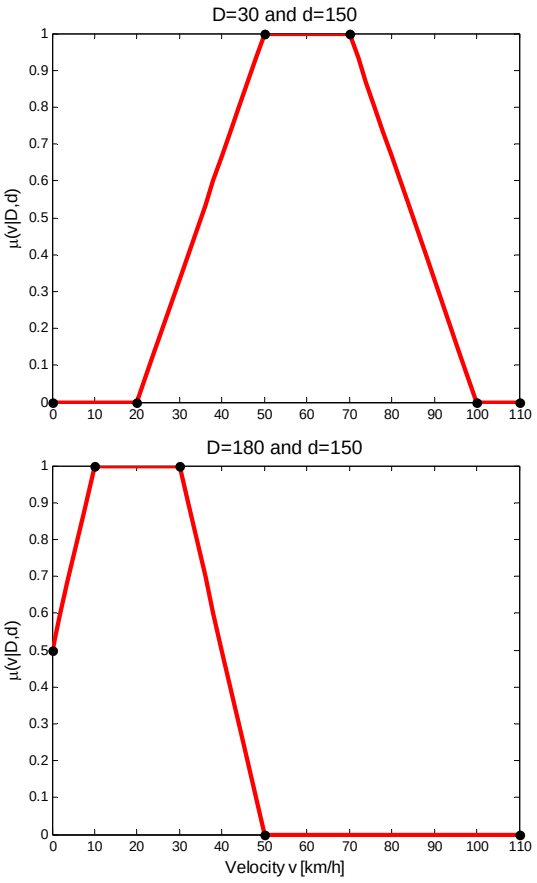


Figure 3: 1D membership functions given a traffic density D = 0 and D = 180 respectively and distance d = 150 with support points (black dots)

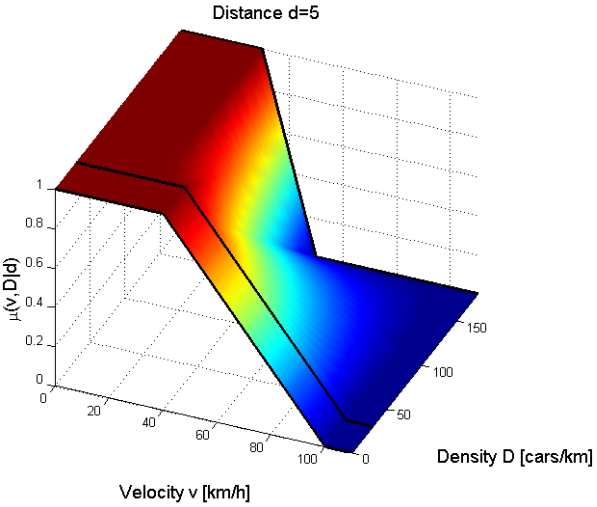


Figure 4: 2D membership function at distance d=5, with 1D support functions (black lines)

To create this 3D function, support points have to be selected. I.e., given an open traffic situation (D = 30) and long distance from an intersection (d = 150) the possibility of a vehicle moving with a speed between 0-20 km/h in shall be 0, while the possibility at the same position in the same traffic situation shall be 1 for velocities between 50-70 and becoming 0 again at v = 100.

A second function depicts a dense traffic situation at the same distance. The possibility for zero velocity shall be 0.5 (since jam cues more likely move slowly forward), while speeds of 10-30 shall be most likely and speeds larger than 50 simply impossible. By linear interpolation between the support points, this results in the functions depicted in Figure 3.

Assuming that the possibilities evolve linearly over the dimension of density, we can derive 2D functions given a certain distance. The function for a position right in front of an intersection is shown in Figure 4.

By linear interpolation along the third axis *d*, we receive a cubic membership function (Figure 5).

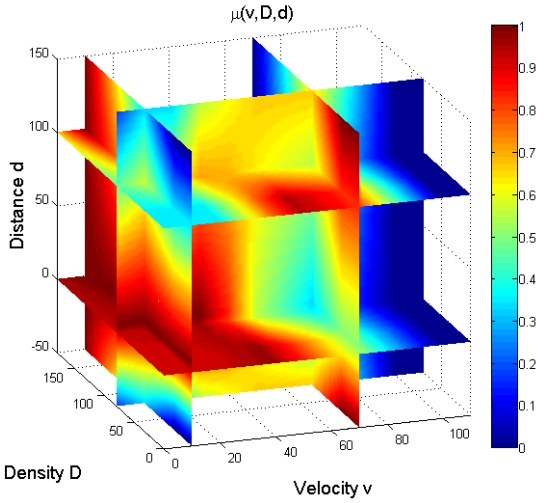


Figure 5: 3D membership function with slices at v=10, v=70, D=80, d=0 and d=100

3.3.2 Evaluation of velocity information

Before the evaluation of the speed, the road is split into sections of length 50m and, near to intersections, of only 20m (sees Figure 6). Every detected object is assigned to one section and contributes to the section density. After the determination of the section density, the possibility μ_A is derived from the above described 3D membership function for each object.

The fuzzy possibility serves a weight in the calculation of a weighted average velocity for each section:

$$\overline{v} = \frac{\sum_i v_i \cdot \mu_A(v_i, D_i, d_i)}{\sum_i \mu_A(v_i, D_i, d_i)} \quad (1)$$

and its standard deviation:

$$\sigma_v^2 = \frac{\sum_i (\overline{v} - v_i)^2 \cdot \mu_A(v_i, D_i, d_i)}{\sum_i \mu_A(v_i, D_i, d_i)} \quad (2)$$

By applying a minimum threshold on the summed up weights of a section, we meet the circumstance that there are only false alarms in a free flow section. If the sum of the weights of a section is below the threshold, all detections of this section are removed.

Finally, objects with a velocity $v_i < \overline{v} - 2 \cdot \sigma_v$ are regarded as outliers and eliminated. Then, a refined and unweighted average velocity is determined from the remaining detections. The resulting distribution is unbiased under the assumption that all false alarms have been eliminated.

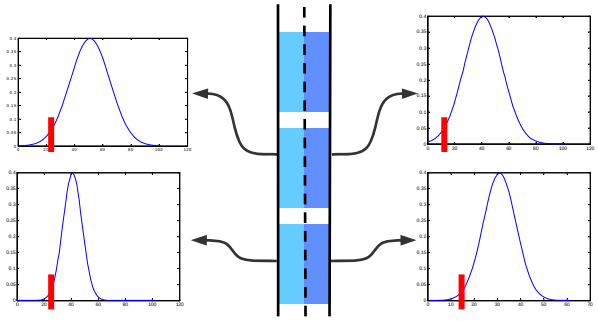


Figure 6: Velocity distribution and cutting-off criteria (red) per road section for each lane

4. RESULTS

The concept has been tested on two different sets of image data so far. The results are shown in Figure 7. One image shows a highway section with free flowing traffic. Here, the detection was carried out by a blob detection algorithm as explained in (Lenhart et al.). In this case, the refinement was able to eliminate all false alarms and redundant objects that arose from the automatic detection.

The second image shows a more complex scene with an urban highway section and an exit leading to an intersection with a traffic light. In this case, the detection has been carried out manually, however, considering a reasonable detection characteristic and quality. In this example, 12 objects have been correctly removed, leaving only one false alarm that could not be eliminated due to faulty tracking.

In both examples, the correctness of the detection could be significantly increased by approximately 30%.

Figure 7: Results of the refinement process. Green: detected and tracked vehicles; Black: redundant objects eliminated by trajectory analysis; Red: false alarms by velocity evaluation. Blue circle: false alarm which has not been eliminated.

5. DISCUSSION AND OUTLOOK

The presented concept shows a possibility to refine detection and tracking results by velocity and trajectory evaluation. This allows a more precise derivation of the average velocity which is fed into traffic models. The benefit of this concept affects exclusively the calculation of the average velocity. It is not possible to counter the problem of low completeness, since missed hits cannot be recovered. However, as has been shown by the simulation in Sect. 2, a low false alarm rate is essential for extrapolating the detection results. For low detection rates, traffic flow parameters can still be estimated with reasonable quality if the false alarm rate is small enough.

Still, many more test scenes have to be evaluated in order to give a more precise measure for the potential of this method.

REFERENCES

Behrisch, M., Bonert, M., Brockfeld, E., Krajzewicz, D., Wagner, P. (2008): Event traffic forecast for metropolitan areas based on microscopic simulation. In: *Third International Symposium of Transport Simulation 2008 (ISTS08)*.

Brackstone, M., McDonald, M. (1999): Car-following: a historical review. In: *Transportation Research Part F: Traffic Psychology and Behaviour*, Volume 2, 181 - 196, 1999.

Ernst, I., Sujew, S., Thiessenhusen, K-U., Hetscher, M., Raßmann, S., Ruhé, M. (2003): LUMOS - Airborne Traffic Monitoring System. In: *Proceedings of the IEEE 6th International Conference on Intelligent Transportation Systems*, Shanghai (China), 12-15 Oct. 2003.

Ernst, I., Hetscher, M. Zuev, S., Thiessenhusen, Kai-Uwe; Ruhé, M. (2005): New approaches for real time traffic data acquisition with airborne systems. In: Stilla, U., Rottensteiner, F., Hinz, S. (eds): *CMRT05*, IAPRS, Vol. 36, Part 3/W24, 69-73.

Everaerts, J., Lewyckyj, N., Fransaer, D. (2004): PEGASUS: Design of a Stratospheric Long Endurance UAV System for Remote Sensing. In: Altan, O.(ed): *Proceedings of the 20th ISPRS Congress*, Istanbul 2004, IAPRS, Vol. 35, Part B2, 29-33.

Hinz, S., Bamler, R., Stilla, U. (2006): *ISPRS Journal Theme Issue: "Airborne and Spaceborne Traffic Monitoring"*. International Journal of Photogrammetry and Remote Sensing, 61(3/4).

Kurz, F., Charmette, B., Suri, S., Rosenbaum, D., Spangler, M., Leonhardt, A., Bachleitner, M., Stätter, R., Reinartz, P. (2007): Automatic Traffic Monitoring with Airborne Wide-Angle Digital Camera System for Estimation of Travel Times. In: Stilla, U., Mayer, H., Rottensteiner, F., Heipke, C., Hinz, S. (eds): *PIA07*, IAPRS, Vol. 36, Part 3/W49A, 83-89.

Rosenbaum, D., Kurz, F., Thomas, U., Suri, S., Reinartz, P. (2008): Towards automatic near real-time traffic monitoring with an airborne wide angle camera system. *European Transport Research Review,* 2008.

Ruhé, M., Hipp, E., Kühne, R. (2007): A model for new data - using air borne traffic flow measurement for traffic forecast. In: *TRISTAN VI* (Sixth Triennial Symposium on Transportation Analysis), Phuket, Thailand, 10-15 June 2007.

Lenhart, D., Hinz, S., Leitloff, J., Stilla, U. (2008): Automatic Traffic Monitoring based on Aerial Image Sequences. In: *Pattern Recognition and Image Analysis, Volume 18, 3*, p. 400-405, Springer, 2008.

Hall, F. (1999): Traffic Stream Characteristics. In: *Traffic Flow Theory - A State-of-the-Art Report*, updated version of Transportation Research Board Special Report 165. Chapter 2. http://www.tfhrc.gov/its/tft/chap2.pdf (accessed 27. March, 2009)

Steger, C. (2001): Similarity measures for occlusion, clutter, and illumination invariant object recognition. In: B. Radig and S. Florczyk (eds.) *Pattern Recognition,* DAGM 2001, LNCS 2191, Springer Verlag, 148–154.

Stilla, U., Rottensteiner, F., Hinz, S.: *CMRT05*, IAPRS, Vol. 36, Part 3/W24.

Zadeh, L. (1965): Fuzzy sets. In: Information and Control, Volume 8, 338 - 353, 1965.

UTILIZATION OF 3D CITY MODELS AND AIRBORNE LASER SCANNING FOR TERRAIN-BASED NAVIGATION OF HELICOPTERS AND UAVs

M. Hebel [a], M. Arens [a], U. Stilla [b]

[a] FGAN-FOM, Research Institute for Optronics and Pattern Recognition, 76275 Ettlingen, Germany - hebel@fom.fgan.de
[b] Photogrammetry and Remote Sensing, Technische Universität München, 80290 München, Germany - stilla@bv.tum.de

KEY WORDS: Airborne laser scanning, LiDAR, GPS/INS, on-line processing, navigation, city models, urban data

ABSTRACT:

Airborne laser scanning (ALS) of urban regions is commonly used as a basis for 3D city modeling. In this process, data acquisition relies highly on the quality of GPS/INS positioning techniques. Typically, the use of differential GPS and high-precision GPS/INS postprocessing methods are essential to achieve the required accuracy that leads to a consistent database. Contrary to that approach, we aim at using an existing georeferenced city model to correct errors of the assumed sensor position, which is measured under non-differential GPS and/or INS drift conditions. Our approach accounts for guidance of helicopters or UAVs over known urban terrain even at night and during frequent loss of GPS signals. We discuss several possible sources of errors in airborne laser scanner systems and their influence on the measured data. A workflow of real-time capable methods for the segmentation of planar surfaces within ALS data is described. Matching planar objects, identified in both the on-line segmentation results and the existing city model, are used to correct absolute errors of the sensor position.

1. INTRODUCTION

1.1 Problem description

Airborne laser scanning usually combines a LiDAR device (light detection and ranging) with high-precision navigational sensors (INS and differential GPS) mounted on an aircraft. Range values are derived from measuring the time-of-flight of single laser pulses, and scanning is performed by one or more deflection mirrors in combination with the forward moving aircraft. The navigational sensors are used to obtain the 3D point associated with each range measurement, resulting in a georeferenced point cloud of the terrain. A good overview and a thorough description of ALS principles can be found in (Wehr & Lohr, 1999). Laser scanning delivers direct 3D measurements independently from natural lighting conditions, and it offers high accuracy and point density.

A well-established application of laser point clouds acquired at urban areas is the generation of 3D city models. However, the overall precision of the derived city model highly depends on the accuracy of the data input, which is directly dependent on the exactitude of the navigational information. Great efforts are usually required during data acquisition and postprocessing in order to achieve high fitting accuracy of multiple ALS datasets (e.g. neighboring strips). While ALS data acquisition is commonly done to supply other fields of studies with the necessary data, few examples can be found where laser scanners are used directly for pilot assistance. One of these examples is the HELLAS obstacle warning system for helicopters (Schulz et al., 2002), which is designed to detect wires and other obstacles for increased safety during helicopter missions.

Despite increasing performance of LiDAR systems, most remote sensing tasks that require on-line data processing are still accomplished by the use of conventional CCD or infrared cameras. Typical examples are airborne monitoring and observation devices that are used for automatic object recognition, situation analysis or real-time change detection. Utilization of these sensors can support law enforcement, firefighting, disaster management, and medical or other emergency services. At the same time, it is often desirable to assist pilots with obstacle avoidance and aircraft guidance in case of poor visibility conditions, during landing operations, or in the event of GPS dropouts. Three-dimensional information as provided by the LiDAR sensor technology can ease these tasks, but the existence of differential GPS ground stations and the feasibility of comprehensive data analysis are not to be considered for these real-time operations.

1.2 Overview

The approach of using ALS information to provide on-line navigation support for aircraft guidance over urban terrain is opposite to the process of city model generation. In contrast to the demand for high-precision positioning techniques, it is assumed that a proper georeferenced city model is already available. This database can be used to generate a synthetic vision of the terrain according to current position and orientation of the aircraft. Moreover, ALS measurements and matching counterparts in the city model can be taken into consideration if additional navigational information is needed, for example in cases of degraded GPS positioning accuracy.

This paper presents a workflow of methods for the segmentation of planar surfaces in ALS data that can be accomplished in line with the data acquisition process. Since most of currently used airborne laser scanners, like the RIEGL LMS-Q560, measure range values in a pattern of parallel scan lines, the analysis of geometric features is performed directly on this scan line data. Straight line segments are first segmented and then connected across consecutive scan lines to result in planar surfaces. All proposed operations are applicable for on-line data processing.

Preparatory work regarding the existing database is required in order to exploit the on-line segmentation results in a pilot assistance system. A set of planar surfaces characterizing the urban terrain is needed for the later comparison (e.g. facades, rooftops). In our experiments, even this information originated from previously collected ALS data, which were recorded under optimal DGPS conditions, but it might as well be derived from any other existing city model.

Within our approach, we assume that INS navigation is continuously available and that we have an initial guess of the sensor position (e.g. from non-differential GPS). If we have to navigate through GPS dropouts, the positioning accuracy will degrade because of INS drift effects, but we can assume that the measured ALS data are still roughly aligned to the stored information. In addition to their position in 3D space, features like size and normal direction are assigned to all segmented planar patches, thus it is comparatively easy to find corresponding objects in the database. We use this information to achieve precise alignment between the measured ALS data and the existing city model, which finally enables us to correct the presumed sensor position.

1.3 Related work

In recent years, airborne laser scanning systems have been explored by various scientists from different points of view. The complexity of ALS data acquisition leads to a number of potential error sources. Schenk (2001) and Filin (2003) address this problem and categorize different influences that should be considered. In addition to varying exactness of the navigational information sources, several systematic effects can lead to reduced point positioning accuracy. Exemplary limiting factors are scanning precision and range resolution of the specific laser scanning device. Other negative effects can be introduced by inaccurate synchronization of the system components. Considerable discrepancies are caused by mounting errors or disregarded lever arms (displacements between laser scanner, INS, and GPS antenna). Skaloud and Lichti (2006) approached this problem with a rigorous method to estimate the system calibration parameters such that 3D points representing a plane are conditioned to show best possible planarity. In order to use ALS within the scope of aircraft navigation, we presuppose that the sensor system has been calibrated beforehand.

Some procedures described in this paper are concerned with the segmentation of point clouds into planar surfaces. Many different methods regarding this topic can be found in literature. Some authors are interested in detecting even more kinds of objects like spheres, cylinders, or cones. Rabbani et al. (2007) describe two methods for registration of point clouds, in which they fit models to the data by analyzing least squares quality measures. Vosselman et al. (2004) use a 3D Hough transform to recognize structures in point clouds. Filin & Pfeifer (2006) propose a segmentation method that is based on cluster analysis in a feature space. Among all available approaches, the RANSAC algorithm (Fischler & Bolles, 1981) has several advantages to utilize in the shape extraction problem (Schnabel et al., 2006). We apply a RANSAC-based robust estimation technique to fit straight line segments to the scan line data. Moreover, an extension of this method is used to identify locally planar patches in the model data. The amount of outliers lets us distinguish between buildings and irregularly shaped objects like trees. Fundamental ideas on fast segmentation of range data into planar regions based on scan line analysis have been published by Jiang and Bunke (1994). Their algorithm divides each row of a range image into straight line segments which are combined in a region growing process. Despite the fact that we are considering continuously recorded scan lines instead of range images, we basically follow this approach during the on-line data analysis.

Several existing concepts of terrain-based navigation for aerial vehicles can be found, e.g. image based navigation (IBN), terrain-following radar (TFR), or terrain contour matching (TERCOM). Other than these methods, laser scanning is a comparatively new technique. Toth et al. (2008) propose the use of LiDAR for terrain navigation, as it provides distinct 3D measurements that can easily be used for exact comparison to previously recorded data. In their concept, the iterative-closest-point algorithm (Besl & McKay, 1992) is chosen for surface matching. Instead of an ICP approach, we identify matching planar objects with regard to several geometric features (i.e. position, size, normal direction). Similar methods have demonstrated high performance for markerless TLS registration (Brenner et al., 2008). The problem of determining the transformation parameters is transferred to a system of linear equations that can be solved immediately.

2. EXPERIMENTAL SETUP

Data used for this study were collected during a field campaign in 2008, using the sensor equipment that is briefly described in this section.

2.1 Sensor carrier

The sensors described below have been attached to a helicopter of type Bell UH-1D (Figure 1). Laser scanner and IMU are mounted on a common sensor platform at the side of the helicopter, which can be tilted to allow different perspectives, i.e. nadir or oblique view. In an operational system, the pilot must be able to react to upcoming dangers, e.g. during degraded visibility conditions. Therefore, an obliquely forward-looking sensor configuration was used in our experiments. The lever arms of the components in the system are known, and the correct bore-sight angles have been determined. Calibration of these parameters is not topic of this paper, suitable methods can be found in (Skaloud & Lichti, 2006).

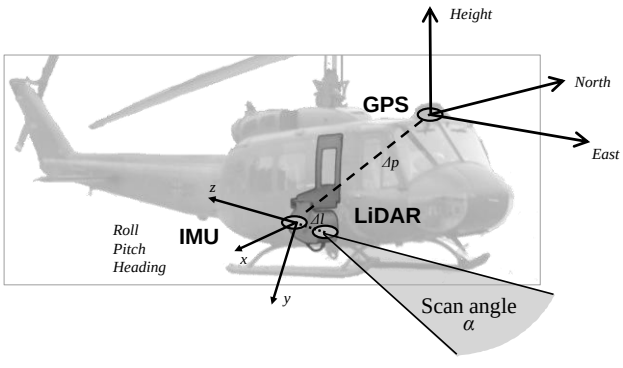


Figure 1. Sensor carrier and sensor configuration.

2.2 Laser Scanner

The RIEGL LMS-Q560 laser scanner makes use of the time-of-flight distance measurement principle with a pulse repetition rate of 100 kHz. Opto-mechanical beam scanning provides single scan lines, where each measured distance can be georeferenced according to position and orientation of the

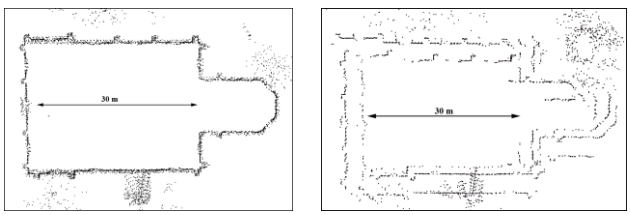
sensor. Waveform analysis can contribute intensity and pulse-width as additional features, but since we are mainly interested in fast acquisition and on-line processing of range measurements, we neglect full waveform analysis throughout this paper. Range d under scan angle α (Figure 1) is estimated corresponding to the first returning echo pulse as it can be found by constant fraction discrimination. Typically, each scan line covers a field of view of 60° with 1000 range measurements (d,α) that can be converted to 2D Cartesian coordinates (Figure 6). Navigational data are synchronously assigned to the range measurements for direct georeferencing.

2.3 Navigational sensor system

The Applanix POS AV 410 comprises a GPS receiver and a gyro-based inertial measurement unit (IMU), which is the core element of the inertial navigation system (INS). The GPS data are used for drift compensation and absolute georeferencing, whereas the IMU determines accelerations with high precision. These data are transferred to the position and orientation computing system (PCS), where they are fused by a Kalman filter, resulting in position and orientation estimates for the sensor platform. In addition to the real-time navigation solution, specialized software can be used for accurate postprocessing of the recorded navigational data. Applanix POSPac MMS incorporates the use of multiple DGPS reference stations and the import of precise GPS ephemeris information. We consider this corrected navigation solution while generating an optimal database of the urban terrain.

3. USED METHODS AND DATA PROCESSING

In this chapter, we distinguish two different operating modes of ALS data acquisition and processing. First, we assume that we have optimal settings for creation of an adequate database: the relevant urban area can be scanned several times from multiple aspects with a calibrated sensor, and data can be processed and optimized off-line. During this stage, we can resort to own differential GPS base stations or use according information, e.g. provided by the "Satellite Positioning Service of the German State Survey" (SAPOS). Under these conditions, the absolute measurement accuracy of an ALS system is typically in the order of one decimeter (Rieger, 2008).



(a) (b)

Figure 2. Horizontal cross-section of a building in overlapping point clouds: (a) after INS/DGPS postprocessing, (b) using the real-time navigation solution.

In the second mode of ALS operation, the data is used for on-line navigation updates during helicopter missions. At this time, we expect non-differential GPS conditions, GPS dropouts, and loss of data points due to smoke, fog, or other negative influences. Figure 2 shows the accuracy that can be obtained in the different operating modes. ALS data in this example were acquired at a skew angle of 45 degree (forward-looking). The helicopter approached the same urban area from six different directions, and the resulting 3D points were combined into a single point cloud. Both illustrations depict the aggregated data within the horizontal cross-section of a building in the overlap area. Best accuracy as shown in Figure 2a results from a global optimization of the navigational data with the Applanix postprocessing software. In this example, the data of six DGPS ground stations were taken into account. Compared with this accuracy, discrepancies of several meters can occur if the real-time navigation solution is used (Figure 2b). This situation will even get worse in case of GPS dropouts.

3.1 Automatic generation of an adequate database

The intended utilization of LiDAR sensors for aircraft guidance does not require a highly detailed GIS. We limit the creation of a database to the extraction of planar patches in multi-aspect ALS point clouds of the relevant urban area. As mentioned before, these data should be collected under optimal conditions (Figure 2a). The combined complete 3D point cloud contains information concerning all facades and rooftops of buildings. A workflow of off-line processing methods is used to filter points and extract most of the planar objects. The respective flowchart is illustrated in Figure 3.

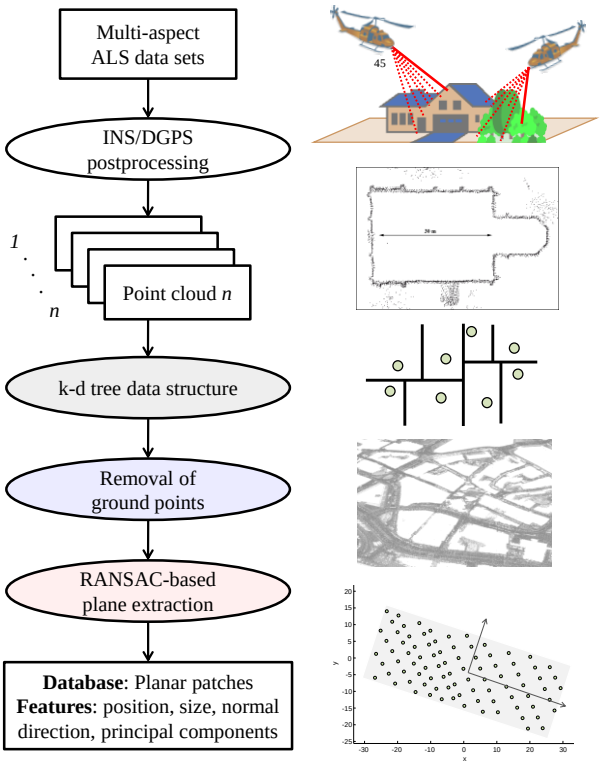


Figure 3. Flowchart of the model creation.

Merging of several multi-aspect ALS data sets results in an irregularly distributed 3D point cloud. We introduce a k-d tree data structure to handle automatic processing of these data. The search for nearest neighbors can be done very efficiently by using the tree properties to quickly eliminate large portions of the search space.

The subsequent segmentation method is intended to keep only those points that are most promising to represent parts of buildings. At first, we remove all ground points by applying a region growing technique in combination with a local analysis of height values. We search for sections of the point cloud in which the histogram of height values clearly shows a

multimodal distribution. There, laser points at ground level appear as the lowest distinct peak. Such positions are then used as seed points for the region growing procedure, which collects all points falling below a certain slope. The necessary search operations are accomplished by means of the k-d tree data structure. In general, this method may misclassify some points (e.g. inner courtyards), but this is negligible for our application. An overview of advanced methods for bare-earth extraction can be found in (Sithole & Vosselman, 2004).

The main step of the model creation is the extraction of planar features from the remaining 3D points. Remarkably, the applied segmentation method is almost identical to the algorithm used for detection of straight line segments in the scan line data, except for the terms line/plane and the different data structure. Similarly, geometric features of the extracted shapes have to be computed in both the model and the on-line results. These topics are described in sections 3.2 and 3.4, respectively. Figure 4 gives an impression of the derived model data. The underlying point cloud is composed of four partial scans of the terrain (different aspects).

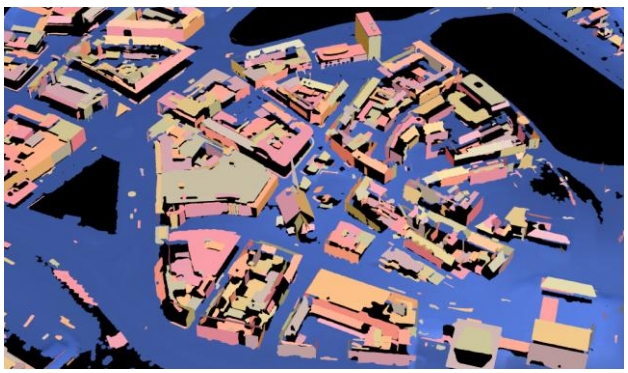


Figure 4. Partial view of the database (model) with ground level (blue) and identified planar patches (red).

3.2 Scan line analysis of airborne LiDAR data

During on-line processing, the analysis of geometric features is performed directly on the scan line data. Most parts of typical buildings will appear as local straight line segments in the 2D Cartesian representation, even if the airborne laser scanner is used in oblique configuration (Figure 1). The RANSAC technique is used to locally fit straight line segments to the scan line data. As mentioned in section 3.1, the algorithm described below can be modified to accomplish segmentation of planar surfaces in a 3D point cloud. This is simply done by replacing the scan line index with the k-d tree data structure, and by turning attention to 3D planes instead of 2D lines.

An overview of the proposed method is shown in Figure 5. Within each iteration step, we randomly select a position in the array A of scan line data points and try to fit a straight line segment to the neighboring data. The RANSAC technique provides a robust estimation of the line segment's parameters. If the fitted straight line is of poor quality, the data associated with the current position is assessed as clutter. Otherwise, we try to optimize the line fitting by looking for all data points that support the previously obtained line, which is done in steps (9), (10), and (11). These steps actually represent a "line growing" algorithm. The local fitting of a straight line segment is repeated with the supporting points to get a more accurate result. The end points of each line segment can be found as the perpendicular feet of the two outermost inliers. Figure 6 shows detected straight lines for an exemplary scan line, depicted with a suitable color-coding according to the normal direction.

(1) Choose an unprocessed position i at random among the available scan line data in the array A.
(2) Check a sufficiently large interval around this position i for available data, resulting in a set S of 2D points.
(3) Set the counter k to zero.

(4) If S contains more than a specific number of points, go to (5). Otherwise mark this position i as discarded and go to step (14).
(5) Increase the counter k by one.
(6) Perform a RANSAC-based straight line fitting with the 2D points in the specified set S.
(7) If the number of inliers is low, mark the current position i as discarded and go to step (14).
(8) Obtain the Hessian normal form $L: (\boldsymbol{x}-\boldsymbol{p})\cdot\boldsymbol{n}_0 = 0$ and push the current position i on a stack (LIFO).

Region growing

(9) Pop the first element j off the stack.
(10) Check each position in an interval around j, which has not already been looked at, whether the respective point lies sufficiently near to the straight line L. If so, push its position on the stack and include the 2D point in a new set S'.
(11) While the stack is not empty, go to (9). Otherwise continue with step (12).

(12) If the counter k has reached its predefined maximum (e.g. two cycles), mark all positions of points in S' as processed and determine the regression line to S'. Store the perpendicular feet of the two outermost points to define the straight line segment and go to step (14). Otherwise continue with (13).
(13) Go to step (4) with the new set of points $S:=S'$.

(14) Repeat from (1) until all points are classified.

Figure 5. Procedure for RANSAC-based shape extraction (example: detection of lines in a set of 2D points).

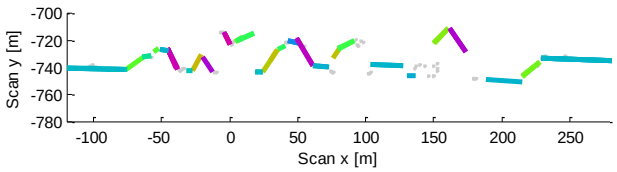


Figure 6. Detected straight line segments in a typical scan line.

3.3 Grouping of line segments

The end points of each line segment are georeferenced to result in correct positioned straight 3D lines. In this section, we describe a procedure to connect coplanar line segments of consecutive scan lines. Let P_i, P_j, and P_k be three of the four end points of two line segments in different scan lines. The distance of the fourth end point P_m to the plane defined by the three others is a measure of coplanarity. We define a distance d_p as the sum of all four possible combinations:

$$d_p := \sum_4 \frac{\left|\left(\boldsymbol{p}_i-\boldsymbol{p}_m\right)^T\left(\boldsymbol{p}_i-\boldsymbol{p}_j\right)\times\left(\boldsymbol{p}_i-\boldsymbol{p}_k\right)\right|}{\left\|\left(\boldsymbol{p}_i-\boldsymbol{p}_j\right)\times\left(\boldsymbol{p}_i-\boldsymbol{p}_k\right)\right\|} \qquad (3.1)$$

The algorithm to find corresponding line segments in a sequence of scan lines can be summarized as follows:

(1) Select the next line segment *a* in the current scan line.

(2) Set the label of *a* to a new and increasing labeling number.

(3) Successively compare line segment *a* to each line segment *b* of several previous scan lines. If Euclidean distances, disparity of normal direction, and the measure of coplanarity d_p are found to be smaller than predefined thresholds, go to step (4). Otherwise go to step (5).

(4) Set the label of *a* to that of *b*.

(5) Continue with (1) until all line segments *a* are processed.

The above steps summarize the main ideas of our method. In fact, we apply an extended two-pass approach to improve detection of connected components. More details on this topic can be found in (Hebel & Stilla, 2008). Figure 7 illustrates the procedure. First, each line segment is initialized with a unique label. Coplanar line segments that are found to lie near to each other are linked together by labeling them with a common labeling number. This process is repeated until all new line segments are labeled. Surfaces are represented by the emerging clusters of line segments with the same label (Figure 8).

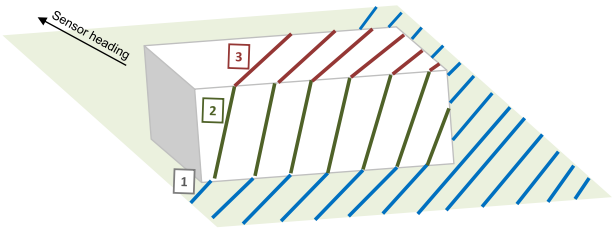


Figure 7. Illustration of scan line grouping.

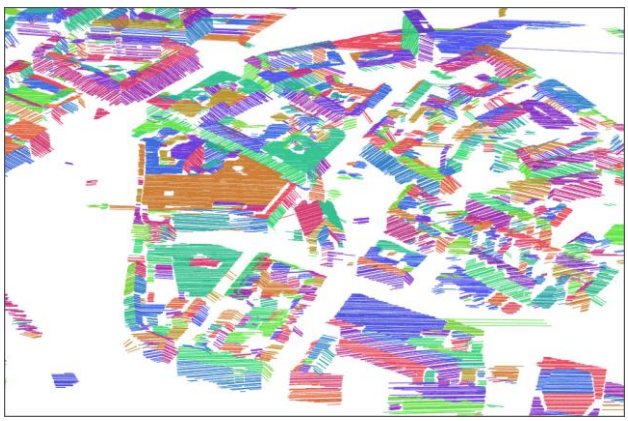


Figure 8. Result of scan line grouping for measured ALS data.

3.4 Feature extraction

Each cluster of connected straight line segments can be characterized by a set of features which are described in this section. For a given cluster of connected line segments, let *C* denote the set of associated 3D data points. The centroid of *C* can be computed as the sum of all points divided by their number, and *C* can be translated towards the origin:

$$\bar{\boldsymbol{c}} = \frac{1}{n}\sum_{i=1}^{n}\boldsymbol{c}_i, \qquad C_0 := C - \bar{\boldsymbol{c}} \tag{3.2}$$

The eigenvectors of the covariance matrix $C_0{}^tC_0$ are the principal components of *C*. The normal direction $\boldsymbol{n}_0$ is given as the normalized eigenvector that corresponds to the smallest eigenvalue. The value of the smallest eigenvalue λ_0 of the covariance matrix, divided by the number of points, is influenced by the curvature and the scatter of *C*. If it is near zero, this indicates a planar surface. The features used to identify matching surfaces in the model data and the results of scan line analysis are: centroid, normal direction, and the normalized eigenvalues of the covariance matrix. These features can even be used to classify and remove irregularly shaped surfaces, e.g. the ground level in Figure 8.

3.5 Registration of ALS and model data

Even without considering terrain-based navigation, we assume that the sensor position is known approximately with standard GPS accuracy. In case of GPS dropouts, the IMU drift will not distort the positioning exactness dramatically. The relative accuracy provided by the INS measurements still ensures consistent ALS measurements over limited periods of time, depending on the quality of the INS system. In some situations, the absolute navigational accuracy needs to be improved. Examples are low-altitude flights of helicopters at night or preparation of landing approaches during rescue missions at urban areas.

If the helicopter is equipped with a LiDAR sensor, ALS data can be collected for several seconds in order to scan the urban area in front of the helicopter (Figure 8). Surfaces that are instantaneously detected in these data can be compared and matched to the existing database of the terrain (Figure 4). The features that are used to establish links have been described in section 3.4. First, the displacement of the centroids has to fall below a maximum distance. Second, the angle between the normal directions should be small (e.g. <5°). Third, the normalized eigenvalues of the covariance matrix $C_0{}^tC_0$ should be similar. Large planes are likely to be subdivided into dissimilar parts, but we are not interested in finding counterparts to all planes. It is sufficient to have some (e.g. 20) correct assignments. Figure 9 illustrates an exemplary pair of associated surfaces. The offset in position and orientation indicates the inaccuracy of the navigational data.

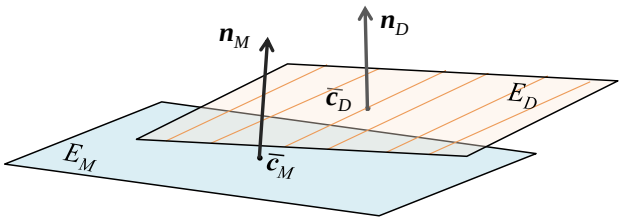


Figure 9. A pair of corresponding planes in model *M* and currently acquired ALS data *D*.

In this section, we determine a rigid transformation $(R,\boldsymbol{t})$ to correct these discrepancies. Let E_M denote the planar surface of the model *M* that is associated with the plane E_D in the currently collected ALS data *D*. The respective Hessian normal form of these planes is given by the centroids and the normal directions $\boldsymbol{n}_M$, $\boldsymbol{n}_D$ (Figure 9). Since both planes should be identical after registration, the centroid of E_D should have zero distance to E_M. Moreover, the two normal vectors should be equivalent if they are normalized to the same half space. In addition to these conditions, we can assume that errors of orientation will not exceed the range of ±5°. That enables us to linearize the equations:

$$\begin{aligned}&\left(R\cdot\bar{\boldsymbol{c}}_D+\boldsymbol{t}-\bar{\boldsymbol{c}}_M\right)\cdot\boldsymbol{n}_M=0\\&\left(R\cdot\boldsymbol{n}_D\right)\cdot\boldsymbol{n}_M=1\end{aligned}\quad,\quad R\approx\begin{pmatrix}1&-\alpha_3&\alpha_2\\\alpha_3&1&-\alpha_1\\-\alpha_2&\alpha_1&1\end{pmatrix}\qquad(3.3)$$

The rotation angles (α_1, α_2, α_3) and the translation components (t_1, t_2, t_3) are the six variables to be determined. Each corresponding pair of planes E_D, E_M yields two linear equations (3.3), therefore at least three pairs have to be identified in the data to compute the rigid transformation (R,$\boldsymbol{t}$). In general, more correspondences can be found at urban areas. The resulting overdetermined system can be solved approximately by inverting the normal equations. In addition, the area of the planar patches can be used as a weighting factor. Finally, the corrected position of the sensor in the model coordinate system is given as $R\cdot\boldsymbol{p}_{\mathrm{GPS}}+\boldsymbol{t}$ and the orientation is corrected to $R\cdot R_{\mathrm{IMU}}$.

4. EXPERIMENTS

We tested the proposed methods on the basis of real sensor data which were recorded 300 meters above the old town of Kiel, Germany. Data available from four flights over this urban terrain led to the database shown in Figure 4. Additional two flights were considered to prove the concept of terrain based navigation (Figure 8). For this purpose, 1000 randomly chosen displacement vectors in the range [5 m, 20 m] were added to the exact sensor positions and it has been checked if these offsets are corrected automatically. Figure 10 shows the average displacement between calculated and exact sensor position against the number of matching pairs of planes. With our data, we were able to reduce the average offset in sensor position to 1.5 m if at least 25 pairs of associated surfaces can be found (standard deviation: 0.5 m). These numbers most likely depend on additional conditions, e.g. aircraft altitude, aircraft speed, number and orientation of facades and rooftops.

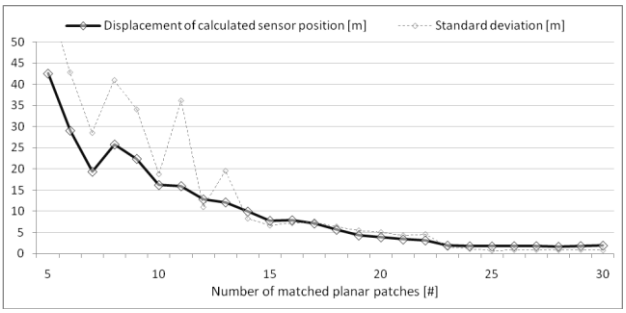


Figure 10. Average displacement against number of planes.

5. CONCLUSION AND FUTURE WORK

The examples presented in this paper were obtained with an experimental sensor system, for which data analysis can only be done offline to show the feasibility of the proposed approach. Nevertheless, we guess that all computations can be accomplished in real-time, with an efficient implementation and appropriate hardware. In our experiments, we were able to align the model and the ALS data such that matching objects show an average distance of 8 cm after the registration. This absolute exactness is not necessarily transferable to the sensor position (see Section 4). With a larger distance between helicopter and the terrain, impreciseness of the sensor orientation has a considerably higher impact on the overall displacement. For example, an angular error of 0.1° would lead to a shift of 1 m in a distance of 600 m. The absolute exactness of the estimated sensor position improves significantly when considering larger areas and/or shorter ranges, e.g. when approaching the terrain at low altitude. In future work, we will analyze these influences in more detail, and we will focus on on-line change detection.

6. REFERENCES

Besl, P.J., McKay, N.D., 1992. A method for registration of 3-D shapes. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 14, No. 2, pp. 239-256.

Brenner, C., Dold, C., Ripperda, N., 2008. Coarse orientation of terrestrial laser scans in urban environments. ISPRS Journal of Photogrammetry and Remote Sensing 63 (1), pp. 4-18.

Filin, S., 2003. Recovery of Systematic Biases in Laser Altimetry Data Using Natural Surfaces. Photogrammetric Engineering & Remote Sensing 69 (11), pp. 1235-1242.

Filin, S., Pfeifer, N., 2006. Segmentation of airborne laser scanning data using a slope adaptive neighborhood. ISPRS Journal of Photogrammetry and Remote Sensing 60 (2), pp. 71-80.

Fischler, M.A., Bolles, R.C., 1981. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. CACM 24 (6), pp. 381-395.

Hebel, M., Stilla, U., 2008. Pre-classification of points and segmentation of urban objects by scan line analysis of airborne LiDAR data. International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences, Vol. 37, Part B3a, pp. 105-110.

Jiang, X., Bunke, H., 1994. Fast Segmentation of Range Images into Planar Regions by Scan Line Grouping. Machine Vision and Applications 7 (2), pp. 115-122.

Rabbani, T., Dijkmann, S., van den Heuvel, F., Vosselman, G., 2007. An integrated approach for modelling and global registration of point clouds. ISPRS Journal of Photogrammetry and Remote Sensing 61 (6), pp. 355-370.

Rieger, P., 2008: The Vienna laser scanning survey. GEOconnexion International Magazine, May 2008, pp. 40-41.

Schenk, T., 2001. Modeling and Analyzing Systematic Errors in Airborne Laser Scanners. Technical Notes in Photogrammetry 19. The Ohio State University, Columbus, USA. 42 p.

Schnabel, R., Wahl, R., Klein, R., 2006. Shape Detection in Point Clouds. Technical report No. CG-2006-2, Universitaet Bonn, ISSN 1610-8892.

Schulz, K.R., Scherbarth, S., Fabry, U., 2002. HELLAS: Obstacle warning system for helicopters. Laser Radar Technology and Applications VII, Proceedings of the International Society for Optical Engineering 4723, pp. 1-8.

Sithole, G., Vosselman, G., 2004. Experimental comparison of filter algorithms for bare-earth extraction from airborne laser scanning point clouds. ISPRS Journal of Photogrammetry and Remote Sensing 59 (1-2), pp. 85-101.

Skaloud, J., Lichti, D., 2006. Rigorous approach to bore-sight self-calibration in airborne laser scanning. ISPRS Journal of Photogrammetry & Remote Sensing 61 (1), pp. 47-59.

Toth, C.K., Grejner-Brzezinska, D.A., Lee, Y.-J., 2008. Recovery of sensor platform trajectory from LiDAR data using reference surfaces. Proceedings of the 13th FIG Symposium and the 4th IAG Symposium, Lisbon, Portugal, 10 p.

Vosselman, G., Gorte, B.G.H., Sithole, G., Rabbani, T., 2004. Recognising structure in laser scanner point clouds. International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences 46 (8), pp. 33-38.

Wehr, A., Lohr, U., 1999. Airborne Laser Scanning – an Introduction and Overview, ISPRS Journal of Photogrammetry and Remote Sensing 54, pp. 68-82.

STUDY OF SIFT DESCRIPTORS FOR IMAGE MATCHING BASED LOCALIZATION IN URBAN STREET VIEW CONTEXT

David Picard[1], Matthieu Cord[1] and Eduardo Valle[2]
[1]LIP6 UPMC
Paris 6
104 avenue du Président Kennedy
75016 Paris FRANCE
{david.picard, matthieu.cord}@lip6.fr
[2]ETIS, CNRS, ENSEA, Univ Cergy-Pontoise,
F-95000 Cergy-Pontoise
mail@eduardovalle.com

KEY WORDS: Image, Databases, Matching, Retrieval, Urban, High resolution

ABSTRACT

In this paper we evaluate the quality of vote-based retrieval using SIFT descriptors in a database of street view photography, a challeging context where the fraction of mismatched descriptors tends to be very high. This work is part of the iTowns project, for which high resolution street views of Paris have been taken. The goal is to retrieve the views of a urban scene given a query picture. We have carried out experiments for several techniques of image matching, including a post-processing step to check the geometric consistency of the results. We have shown that the efficiency of SIFT based matching depends largely on the image database content, and that the post-processing step is essential to the retrieval performances.

1 INTRODUCTION

In this paper, we evaluate the effectiveness of a voting strategy using SIFT descriptors for near-duplicate retrieval of urban scenes. We have observed that, compared to previously repported applications of SIFT (object recognition, stereoscopy, etc.) (Lowe, 2003) this context presents the challenge of a very high rate of descriptor mismatches, due to the complexity of both the scene and the transformations it might suffer. We have thus, evaluated how different strategies to filter out the false matches can improve the effectiveness of retrieval.

This study is part of the iTowns project, which is about defining a new generation of multimedia web tools that mixes a broadband 3D geographic image-based browser with an image-based search engine [1]. Fig. 1 shows an example of pictures taken for the project.

The first goal of the new type of search engine, is to retrieve, in the high-resolution database, the scene corresponding to a given query image. Let us imagine the following scenario: a user is looking for information about a restaurant in front of him (feedback from patrons, for instance). He takes a picture of the restaurant with his phone and send it to the iTowns web server. The image is matched on the database and the desired information is retrieved and sent back to the user.

In order to accomplish this goal, there is basically three steps to perform :

1. Match the query image with the corresponding scene in the database.
2. Find information associated with the scene and related to the query.
3. Retrieve only relevant information regarding the user interests.

[1]See http://itowns.ign.fr

In this paper, we focus on the first part, and consider the use of state of the art techniques for near-duplicate image matching. Recently, techniques have been developed for the detection of copies where transformations between images are well known (rotation, scaling, global illumination change etc). Those techniques involve the extraction of points of interest in the images, then the matching of the points in the query with the points in the database, and the aggregation of the matches for images of the database using a voting strategy. We try to extend these techniques to the matching of images with less constrainted, and thus more realistic transformations (change of viewpoint, local illumination, etc).

The paper is organized as follows: the next section introduces keypoint-based image matching. We explain in section 3 the strategy used to perform an efficient approximate *k-NN* search in the database in order to associate query points with points in the database. Then, we detail in section 4 the geometrical consistency used to filter irrelevant matches. Experiments are done on two representative subsets of the iTowns collection, and results are shows in section 5, before we conclude.

2 KEYPOINTS BASED IMAGE MATCHING

The essential elements of keypoint-based image matching appeared in (Schmid and Mohr, 1997): the use of points of interest, local descriptors computed around those points, a dissimilarity criterion based on a vote-counting algorithm, and a step of consistency checking on the matches before the final vote count and ranking of the results. We use the SIFT points of interest (Lowe, 2003) to describe the

Figure 1: Panoramic view of the *Place de la Nation* from the project iTowns.

image (Fig. 2). The SIFT descriptor consists in a 128-dimensional vector containing a set of gradient orientation histograms.

Figure 2: SIFT points of interest with respecting scales.

The classic method to use keypoints for image matching is pair-wise image comparison. For all points of a query image A, find the best matching point in a target image B. If the resulting match has good contrast (*i.e.* the distance of the query point to the best point in B is far less than the distance to the second best, meaning that the query point has only one corresponding target point), add a vote to B. An example of matching points is shown on Fig. 3. The best image in the database corresponding to the query image is the one with higher votes.

Figure 3: SIFT points matching between a query taken with a mobile phone and an image from the iTowns database.

One of the problems of pair-wise image comparison is that it induces a sequential, linear-time, processing, which is unfeasible for large databases. Hence, instead of finding best matches between keypoints of query and target images, the best matches are found between the query and the keypoints in the entire collection. The retrieval scheme is as follows :

1. For each points in the query, find the *k-nearest neighbours* (*k-NN*) in the database.
2. For each neighbour found, add one vote to the corresponding image.
3. Rank image by descending number of votes.

The main difference with pair-wise comparison is that each keypoint of the query has k associated matches. Thus, points of the query with no corresponding points in the database (points of objects that are not in the database for instance) will still vote. Those votes are randomly distributed among images and thus contribute to increase the ranking of irrelevant images.

In order to remove the influence of those irrelevant points, a geometrical constraint is applied to the matches, removing points in the target that are not coherent with the spatial distribution of points in the query.

3 APPROXIMATE K-NN SEARCH

There are several techniques for efficient kNN search on large databases, like the KD-tree (Friedman et al., 1976), the LSH (Indyk and Motwani, 1998) or projective methods (Kleinberg, 1997). A comprehensive study can be found in (Valle, 2008). Those methods are all approximate because, in order to obtain more efficiency they sacrifice exactness in the name of speed. That means that they find the correct answers with good probability, but not certitude.

We have chosen Multicurves (Valle et al., 2008), a method based on space-filling curves, which are fractal curves able to map the dimensions of the input space into an one-dimensional space, while locally preserving the order (*i.e.*, putting near in the curve point which are near in the space). The one-dimensional data can then be indexed using traditional efficient techniques.

The particularity of Multicurves is using several of those curves at once: first, it projects the input space into a few moderate-dimensional subspaces, then it uses one space filling curve to index each one of those subspaces. This allows the method to better manage the problems associated to high-dimensional indexing. In our experiments, we have used Multicurves with 4 curves, each of them indexing 32 of the 128 dimensions that compose the SIFT input

space. Details of the method as well as its evualuation for copy detection can be found in (Valle et al., 2008).

Each keypoints of the database within the *k-NN* is added to the list of matches of the image it belongs. A basic method to retrieve images corresponding to the scene is to rank the images by descending order of the size of the lists of matches.

4 GEOMETRICAL CONSISTENCY

Since every point of the query is associated with many points in the database, irrelevant points of the query will still influence the ranking. However, we can make the assumption that for those matches, the relative positions of the query and target points within their respective images are not coherent.Thus, a geometric constraint over the ensemble of matches between two images shall be able to remove the irrelevant matches.

We test two criteria of geometrical consistency. The first criteria is to estimate the 2D affine transform between the two images, and then to remove the points not coherent with it. Although the transformation between the images is indeed 3D, we assume that under small perspective changes, a 2D affine transform is enough to catch the transformation of a single plane (in our case, the front of the building). The algorithm used to estimate the affine transform is RANSAC, a model estimation technique which can deal with a large fraction of outliers (Fischler and Bolles, 1981). An example of matches after the removal of non-coherent points is shown on Fig. 4.

Figure 4: Matching points after the non-coherent to the estimate 2D affine transform matches have been remove.

The second criterion is to keep only the matches which correspond to the most frequent angle difference between matched points (Jegou et al., 2008). This is done by computing an histogram of the difference between the principal direction of the query and the target point of a match. We then keep the matches corresponding to the most frequent value in the histogram.

5 EXPERIMENTS

5.1 Protocol

We have tested four methods for comparison on two subset of the *iTowns* images, namely:

- A pairwise matching using a distance contrast criterion (named *Image Matching* there after).
- A *k-NN* search plus a vote (named *Brute vote*).
- A *k-NN* search plus the angle differences consistency criterion (named *Angle differences*).
- A *k-NN* search plus the 2D affine transform consistency criterion (name *Ransac*).

We set $k = 10$ for the *k-NN* Search. The parameters for the *RANSAC* algorithm were empiricaly set to 15 pixels maximum distance to fit the model and minimum 3 inliers for the affine transformation.

The first dataset consisted of 82 images of a single street (about 350 000 keypoints). The query set contained images taken by a mobile phone in front of some of the shops in the street. As the images (both in the query set and in the database) are direct views of the buildings, we considered this test as easy, since the transformation between query and its corresponding target images is simple. The second dataset contained 300 images of a large boulevard (about 3.5 millions of keypoints). The queries where taken with a mobile phone from the sidewalk. As the vehicle taking the pictures was in the middle of the street, the targeted regions of the images (a shop, for instance) are very small. Thus, few keypoints of each image are describing something we might be looking for. As there are many severe transformations (scaling, viewpoint changes), we consider this test difficult. For both sets, we have manually built the groundtruth by annotating which images correspond to each query.

We have used three criteria for the evaluation. The first consisted in measuring the rank of the first relevant image retrieved (average of the query set). The second measure was the evolution of the number of relevant image in the retrieved set, as the size of this set increased. The third criterion was the precision, the number of relevant images retrieved over the number of images retrieved.

5.2 Results on Dataset 1

An example of the first images retrieved using the *Brute vote* is shown on Fig. 5. The first images retrieved with this technique have about 2000 matching keypoints (images in this set contain about 5000 keypoints). There are several occlusions between the query image and the images of iTowns (for instance the car in front of the shop). However, a relevant image is found within the first images.

Fig. 6 presents the same result, but with the *Angle differences* refinement. The first images retrieved have about 200 matching keypoints. As we can see, the refinement introduced a re-ranking of the first images profitable to

Figure 5: First images retrieved using *Brute vote*. The query has a dark red border, while relevant images have a bright green border.

the relevant image. The same query but with the *Ransac* method is shown on Fig. 7. Images retrieved have less than 10 matching keypoints. The removal of non-coherent points increases the ranks of relevant images. The improvement is thus better than the one of the *Angle differences* refinement.

Figure 6: First images retrieved using the *Angle differences* refinement.

Figure 7: First images retrieved using the *Ransac* refinement.

We have computed the mean best rank among relevant images for a set of ten queries. We also compared the multicurves approach to a linear processing of the database for the *k-NN* search, in order to see the influence of the approximate search. The ranks and times are shown in Table 1.

Method	mean best rank	time
Image matching	27.09	11514s
Linear search	5.45	22967s
Brute vote	14	447s
Ransac	**1.09**	-
Angle Differences	7.91	-

Table 1: Mean best rank for the first dataset. '-' denotes a time not computed.

As we can see, the time used for the pair-wise comparison or for the linear *k-NN* search are prohibitive. Since *Brute vote* uses Multicurves, which is an approximate *k-NN* method, we should expect some degradation when compared to *Linear search*, which uses the costly exact *k-NN* search. We note, however, that by using *Ransac*, the precision lost is more than compensated. The *Ransac* refinement has the best results, and is totally satisfactory from the users point of view.

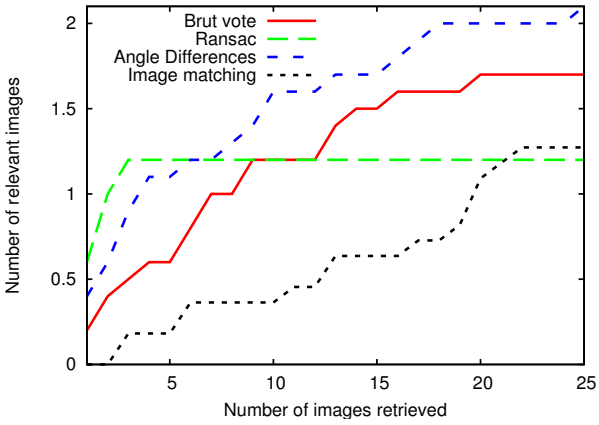


Figure 8: Evolution of the number of relevant images against the number of images retrieved.

We measure the evolution of the number of relevant images as the percentage of the database retrieved increases on Fig. 8. The *Ransac* method outperforms the other in the beginning of the retrieval, but then stops to retrieve images (if no coherent affine transform is found, then the image has a null vote). The *Angle Differences* and the brute voting are less efficient, but still manage to retrieve relevant images within the top 10 images. The pair-wise comparison fails to showing relevant images within the top 10.

The precision (ratio between number of relevant images retrieved and total images retrieved) is shown on Fig. 9. The precision within the first five images retrieved (which is the most relevant metric to the user) is better for the *Ransac* refinement. Past this point, all three *k-NN* based methods are almost equivalent. The pair-wise comparison is surprisingly worse than the other methods.

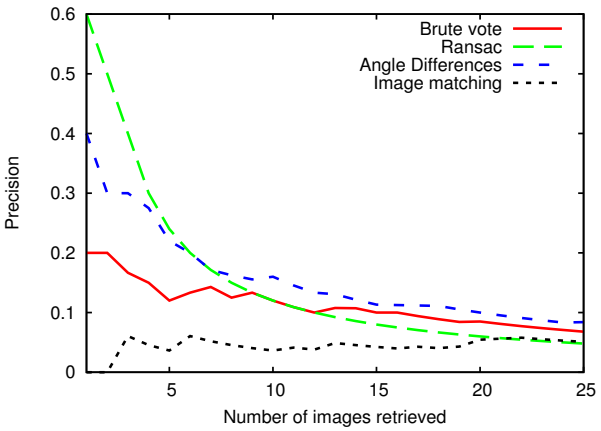


Figure 9: Evolution of the precision against the number of images retrieved.

5.3 Results on dataset 2

An example of results using the brute voting is shown on fig Fig. 10. As we can see, none of the top images are relevant. The same occurs with the angle differences refinement.

Figure 10: Example of first images retrieved using the *k-NN* voting for the second subset.

The RANSAC refinement (Fig. 11) is able to retrieve two relevant images within the first five images, which means that irrelevant matches have been well filtered out.

Like we did for the first subset, we compute the mean best rank shown in table 2. We were not able to compare with linear *k-NN* search due to the time taken by this method.

The first observation is that none of the methods is able to retrieve even one relevant image within the top ten, which means that the methods are not able to give satisfying results from the users point of view. Nevertheless, the geometric consistency step (either *Ransac* or the *Angle Differences*) provides a nice improvement.

Figure 11: Example of first images retrieved using the *Ransac* refinement for the second subset.

Method	mean best rank
Image matching	80.67
Brute vote	98.80
Ransac	**34.40**
Angle Differences	59.10

Table 2: Mean best rank for the second dataset.

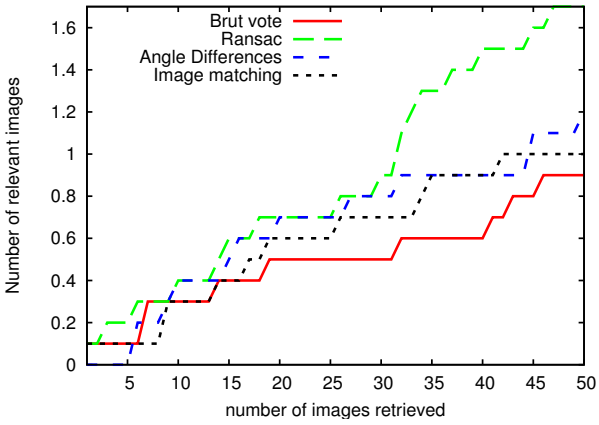


Figure 12: Evolution of the number of relevant images against the number of images retrieved.

The evolution of the number of relevant images is shown on Fig. 12. As we can see, all methods are almost equivalent, with the *Ransac* strategy being a little better for the last 20 images of the top 50.

The precision is shown on Fig. 13, and is very low for all methods. The best result is obtained for the *Ransac*

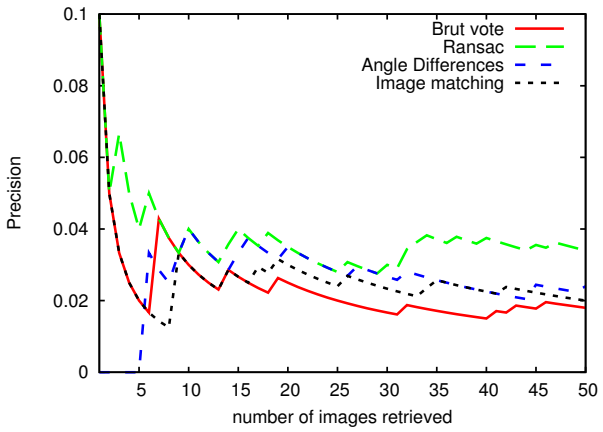


Figure 13: Evolution of the precision against the number of images retrieved.

strategy, but it is still under 5% most of the retrieval. In overall, all methods failed at finding the relevant scene in the database.

6 CONCLUSION

In this paper, we have reviewed the use of keypoints based voting strategy for image matching in the context of the iTowns project. We have tested different strategies (pair-wise comparison, *k-NN* search with brute voting, angle differences refinement, and 2D affine transform estimation) on two subset of urban scene database.

We have first found that there is no penalty in using an approximate *k-NN* search, which is a huge improvement on the retrieval speed. Even for small datasets like the first we used, a pair-wise comparison or a linear *k-NN* search is not feasible for interactive application.

The second point we have found is that the post-processing of the voting strategies is essential to the success of the retrieval. The *Ransac* refinement is the only one able to retrieve at least one relevant image within the first five images, which is the main criterion for a user in this kind of task. A further improvement could be the estimation of more complexe transformation that are more robust to perspective changes.

However, overall results largely depend on the database content. In the case of a small database (which can be obtained through geolocalization) with well taken pictures like the first we used, the results are good enough to be used in the intended application.For the second database, the quality of the results is very low, making them inadequate for our applications. This lack of quality might be an intrinsic characteristic of SIFT when confronted to images like ours, that contain many problematic features (complex shadows, trees, branches, etc), which spawn a huge amount of descriptors with low discriminant power. Those points increase dramatically the number of false matches, inflating the rank of of non relevant images (such as on Fig. 14, which has more matches than the relevant images). As improvement, we suggest a filtering of the database in order to remove points that are not informative.

To conclude, we consider the extension of keypoints based method from copy detection to the matching of scene in difficult context as not successful. We think there is more work to do both on the descriptors and on the matching process. We intend to share our databases and groundtruth with the community in order to allow the benchmarking of those tasks on difficult images.

Figure 14: False matching between two images after geometric consistency check.

REFERENCES

Fischler, M. A. and Bolles, R. C., 1981. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Commun. ACM 24(6), pp. 381–395.

Friedman, J., Bentley, J. L. and Finkel, R. A., 1976. An algroithm for finding best matches in logarithmic expected time. Technical report, Stanford, CA, USA.

Indyk, P. and Motwani, R., 1998. Approximate nearest neighbors: towards removing the curse of dimensionality. In: STOC '98: Proceedings of the thirtieth annual ACM symposium on Theory of computing, ACM, New York, NY, USA, pp. 604–613.

Jegou, H., Douze, M. and Schmid, C., 2008. Hamming embedding and weak geometric consistency for large scale image search. In: A. Z. David Forsyth, Philip Torr (ed.), European Conference on Computer Vision, LNCS, Vol. I, Springer, pp. 304–317.

Kleinberg, J. M., 1997. Two algorithms for nearest-neighbor search in high dimensions. In: STOC '97: Proceedings of the twenty-ninth annual ACM symposium on Theory of computing, ACM, New York, NY, USA, pp. 599–608.

Lowe, D., 2003. Distinctive image features from scale-invariant keypoints. In: International Journal of Computer Vision, Vol. 20, pp. 91–110.

Schmid, C. and Mohr, R., 1997. Local grayvalue invariants for image retrieval. IEEE Trans. Pattern Anal. Mach. Intell. 19(5), pp. 530–535.

Valle, E., 2008. Local-Descriptor Matching for Image Identification Systems. PhD thesis, Univ. Cergy-Pontoise, ETIS, UMR CNRS 8051. Direction : S. Philipp-Foliguet, M. Cord.

Valle, E., Cord, M. and Philipp-Foliguet, S., 2008. High-dimensional descriptor indexing for large multimedia databases. In: CIKM '08: Proceeding of the 17th ACM conference on Information and knowledge management, ACM, New York, NY, USA, pp. 739–748.

TEXT EXTRACTION FROM STREET LEVEL IMAGES

J. Fabrizio[1,2], M. Cord[1], B. Marcotegui[2]
[1]UPMC Univ Paris 06
Laboratoire d'informatique de Paris 6, 75016 Paris, France
[2]MINES Paristech, CMM- Centre de morphologie mathématique, Mathématiques et Systèmes,
35 rue Saint Honoré - 77305 Fontainebleau cedex, France

KEY WORDS: Urban, Text, Extraction, Localization, Detection, Learning, Classification

ABSTRACT

We offer in this article, a method for text extraction in images issued from city scenes. This method is used in the French iTowns project (iTowns ANR project, 2008) to automatically enhance cartographic database by extracting text from geolocalized pictures of town streets. This task is difficult as 1. text in this environment varies in shape, size, color, orientation... 2. pictures may be blurred, as they are taken from a moving vehicle, and text may have perspective deformations, 3. all pictures are taken outside with various objects that can lead to false positives and in unconstrained conditions (especially light varies from one picture to the other). Then, we can not make the assumption on searched text. The only supposition is that text is not *handwritten*. Our process is based on two main steps: a new segmentation method based on morphological operator and a classification step based on a combination of multiple SVM classifiers. The description of our process is given in this article. The efficiency of each step is measured and the global scheme is illustrated on an example.

1 INTRODUCTION

Automatic text localization in images is a major task in computer vision. Applications of this task are various (automatic image indexing, visual impaired people assistance or optical character reading...). Our work deals with text localization and extraction from images in an urban environment and is a part of iTowns project (iTowns ANR project, 2008). This project has two main goals : 1. allowing a user to navigate freely within the image flow of a city, 2. Extracting features automatically from this image flow to automatically enhance cartographic databases and to allow the user to make high level queries on them (go to a given address, generate relevant hybrid text-image navigation maps (itinerary), find the location of an orphan image, select the images that contain an object, etc.). To achieve this work, geolocalized set of pictures are taken every meter. All images are processed off line to extract as many semantic data as possible and cartographic databases are enhanced with these data. At the same time, each mosaic of pictures is assembled into a complete immersive panorama (Figure 1).
Many studies focus on text detection and localization in images. However, most of them are specific to a constrained context such as automatic localization of postal addresses on envelopes (Palumbo et al., 1992), license plate localization (Arth et al., 2007), text extraction in video sequences (Wolf et al., 2002), automatic forms reading (Kavallieratou et al., 2001) and more generally "documents" (Wahl et al., 1982). In such context, strong hypothesis may be asserted (blocks of text, alignments, temporal redundancy for video sequences...). In our context (*natural scenes* in an urban environment), text comes from various sources (road sign, storefront, advertisements...). Its extraction is difficult: no hypothesis can be made on text (style, position, orientation, lighting, perspective deformations...) and the amount of data is huge. Today, we work on 1 TB for a part of a single district in Paris. Next year, more districts will be processed (more than 4 TB). Different approaches already exist for text localization in natural scenes. States of the art are found in (Mancas-Thillou, 2006, Retornaz and Marcotegui, 2007, Jung et al., 2004, Jian Liang et al., 2005). Even if preliminary works exist in natural scene (Retornaz and Marcotegui, 2007, Chen and Yuille, 2004), no standard solution really emerges and they do not focus on urban context.
The paper presents our method and is organized as follows: the text localization process is presented and every step is detailed followed by the evaluation of main steps. In the last part, results are presented. Then comes the conclusion.

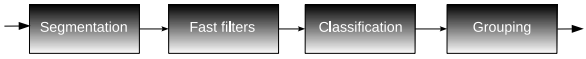


Figure 2: General principle of our system.

2 SEGMENTATION BASED STRATEGY

The goal of our system is to localize text. Once the localization is performed, the text recognition is carried out by an external O.C.R. (but the system may improve the quality of the region by correcting perspective deformations for example). Our system is a region based approach and starts by isolating letters, then groups them to restore words and text zones. Region based approach seems to be more efficient, such approach was ranked first (Retornaz and Marcotegui, 2007) during ImagEval campaign (ImagEval, 2006). Our process is composed of a cascade of filters (Figure 2). It segments the image. Each region is analysed to determine whether the region corresponds to text or not. First stages during selection eliminate a part of non text regions but try to keep as many text region as possible (at the price of a lot of false positives). At the end, detected regions that are close to other text regions are grouped all together. Isolated text regions are canceled.

Figure 1: Image from iTowns project.

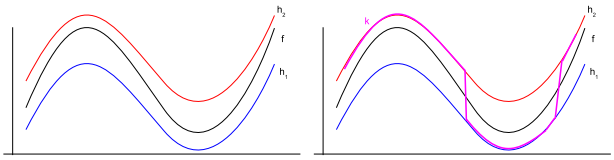


Figure 3: On the left, function f and a set of 2 functions h_1 and h_2. On the right, function k computed by toggle mapping.

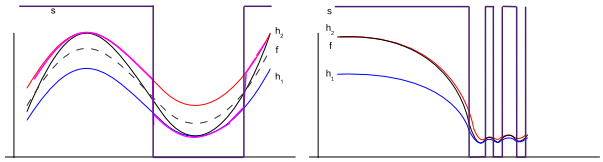


Figure 4: Result of eq. 4 (function s) on an edge and in homogeneous noisy regions.

Figure 5: From left to right: 1. Original image, 2. Binarization (function s from eq. 4), 3. Homogeneity constraint (eq. 5), 4. Filling in small homogeneous regions.

3 TEXT SEGMENTATION

Our segmentation step is based on a morphological operator introduced by Serra (Serra, 1989): *Toggle Mapping*. Toggle mapping is a generic operator which maps a function on a set of n functions: given a function f (defined on D_f) and a set of n functions $h_1, ..., h_n$, this operator defines a new function k by (Fig. 3):

$$\begin{aligned} \forall x \in D_f \; k(x) \quad &= \quad h_i(x); \forall j \in \{1..n\} \\ &\quad |f(x) - h_i(x)| \leq |f(x) - h_j(x)| \end{aligned} \tag{1}$$

The result depends on the choice of the set of functions h_i. A classical use of toggle mapping is contrast enhancement: this is achieved by applying toggle mapping on an initial function f (an image) and a set of 2 functions h_1 and h_2 extensive and anti-extensive respectively.

To segment a gray scale image f by the use of toggle mapping, we use a set of 2 functions h_1 and h_2 with h_1 the morphological erosion of f and h_2 the morphological dilatation of f. These two functions are computed by:

$$\forall x \in D_f \quad h_1(x) = \min f(y); y \in v(x) \tag{2}$$

$$\forall x \in D_f \quad h_2(x) = \max f(y); y \in v(x) \tag{3}$$

with $v(x)$ a small neighborhood (the structuring element) of pixel x. Then, instead of taking the result of toggle mapping k (eq. 1), we keep the number of the function on which we map the pixel. This leads us to define function s:

$$\forall x \in D_f s(x) = i; \forall j \in \{1..2\} |f(x) - h_i(x)| \leq |f(x) - h_j(x)| \tag{4}$$

Function $s(x)$ takes two values and may be seen as a binarization of image f with a local criterion (Fig. 4 left). Our function efficiently detects boundaries but may generate salt and pepper noise in homogeneous regions (Fig. 4 right): even very small local variations generate an edge. To avoid this, we introduce a minimal contrast c_{min} and if $|h_1(x) - h_2(x)| < c_{min}$, we do not analyse the pixel x.

Function s is then improved:

$$s(x) = \begin{cases} 0 & if |h_1(x) - h_2(x)| < c_{min} \\ 1 & if |h_1(x) - h_2(x)| >= c_{min} \\ & \& \; |h_1(x) - f(x)| < p * |h_2(x) - f(x)| \\ 2 & otherwise \end{cases} \tag{5}$$

Then, no boundary will be extracted within homogeneous areas. s is a segmentation of f (notice that now we have 3 possible values instead of 2: a low value, a high value and a value that represents homogeneous regions).

To use this method efficiently, some parameters must be set up: the size of the structuring element used to compute a morphological erosion (h_1) and a dilation (h_2), the minimal contrast c_{min} and an additional parameter p. Variations of p influence the thickness of detected structures.

Getting three values in output instead of two can be disturbing. Many strategies can be applied to assign a value to homogeneous regions (to determine whether the region belongs to low value areas or high value ones): if a region is completely surrounded by pixels of the same value, the whole region is assigned to this value. Another strategy consists in dilating all boundaries onto homogeneous regions. In our case, this is not a real issue: as characters are narrow, it is not common to have homogeneous regions inside characters and if it occurs, such regions are small. Then, our strategy consists in studying boundaries of small homogeneous regions in order to fill a possible hole in characters. Bigger homogeneous regions are mostly left unchanged, only a small dilation of these boundaries is performed.

Illustration of the segmentation process is given in Figure 5. In the rest of the paper, this method is called Toggle Mapping Morphological Segmentation (TMMS).

4 FILTERING

Once the image is segmented, the system must be able to select which regions contain text (letters) and which do not. A part of these regions is obviously non text (too big/too small regions, too large...). The aim of this step is to dismiss most of these obviously non text regions without loosing any good character. A small collection of fast filter (criteria opening) eliminate some regions with simple geometric criteria (based on area, width and height). These simple filters help saving time because they rapidly eliminate many regions, simplifying the rest of the process (which is a bit slower).

5 PATTERN CLASSIFICATION

Some segmented regions are dismissed by previous filters but a lot of false positives remain. To go further, we use classifiers with suitable descriptors.
Due to the variability of analysed regions, descriptors must (at least) be invariant to rotation and scale. The size and the variability of examples in training database ensure to be invariant to perspective deformations. We have tested a lot of different shape descriptors (such as Hu moments, Fourier moments...). Among them, we have selected two families of moments : Fourier moments and the pseudo zernike moments. We select them empirically as during our test, they get a better discrimination ratio than others. We choose also to work with a third family of descriptors: polar representation is known to be efficient (Szumilas, 2008) but the way this representation is used does not match our need. Then we define our own polar descriptors: the analysed region is expressed into polar coordinate space centered into the gravity center (Figure 6). The feature is then mapped into a normalized rectangle (the representation is then invariant in scale factor). To be rotation invariant, many people use this representation by computing a horizontal histogram within this rectangle but this leads to a loss of too much information. Another way to be rotation invariant if the representation used is not rotation invariant is to redefine the distance computed between samples (Szumilas, 2008). But this leads to a higher complexity. To be rotation invariant, we simply take the spectrum magnitude of Fourier transform of each line in the normalized rectangle. These results carry much more information than simple histograms, and are easier than changing the distance used.

Once we choose the descriptors, we train a svm classifier (Cortes and Vapnik, 1995) for each family of descriptors. To give a final decision, all outputs of svm classifier are processed by a third svm classifier (Figure 7). We tried to add more classifiers in the first step of the configuration (with other kinds of descriptors) but this makes the overall accuracy systematically decreasing.

6 GROUPING

We are able to analyse main regions in the image and extract characters. Once these characters are selected, they

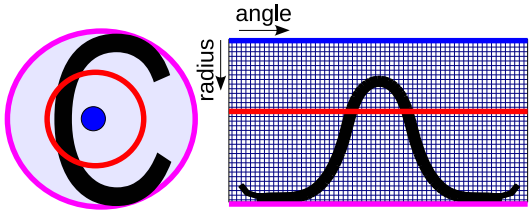


Figure 6: The region is expressed in a polar coordinate space and to have a rotation invariant descriptor we take the spectrum of Fourier transform of every line.

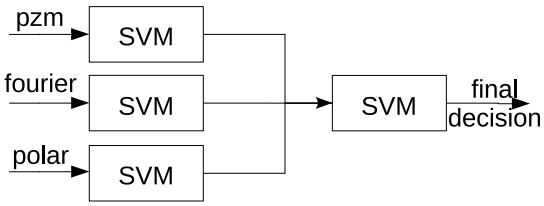


Figure 7: Our classifier is composed of 3 svm classifiers that use common family of descriptors and a svm that take the final decision.

are grouped all together with neighbour to recover text regions. The conditions to link two characters to each other are the one given in (Retornaz and Marcotegui, 2007). They are based on the distance between the two regions relatively to their height. This steps will soon be improved to handle text in every direction as this approach is restricted to nearly horizontal text. During this process, isolated text regions (single character of couple of letters) are dismissed. This aggregation is mandatory to generate words and sentences to integrate as an input in an O.C.R. but it also suppresses a lot of false positive detections.

7 LETTER DETECTION EXPERIMENTS

In this section, we evaluate segmentation and classification steps.

Segmentation The segmentation evaluation is always difficult as it is, for a part, subjective. Most of time, it is impossible to have a ground truth to be used with a representative measure. To evaluate segmentation as objectively as possible for our application, we have constituted a test image database by randomly taking a subset of the image database provided by I.G.N. (Institut Géographique National, n.d.) to the project (iTowns ANR project, 2008). We segment all images from this database and we count properly segmented characters. We define as clearly as possible what *properly segmented* means: the character must be readable, it must not be split or linked with other features around it. The thickness may vary a little provided that its shape remains correct. We compare the result with 3 other segmentation methods:

- Niblack binarization criterion (Niblack, 1986) which

evaluates a threshold $T(x)$ for a given pixel x, according to its neighborhood by:

$$T(x) = m(x) + ks(x) \quad (6)$$

with m and s the mean and the standard deviation computed on the neighborhood and $k \in \mathbf{R}$ a parameter.

- Sauvola binarization criterion (Sauvola et al., 1997) which evaluates a threshold $T(x)$ by:

$$T(x) = m(x)\left(1 + k\left(\frac{s(x)}{R} - 1\right)\right) \quad (7)$$

with R the dynamic of standard deviation $s(x)$.

- the segmentation exposed by Retornaz (Retornaz and Marcotegui, 2007) based on the *ultimate opening*. This operator, introduced by Beucher (Beucher, 2007), is a non-parametric morphological operator that highlights the most contrasted areas in an image.

The evaluation image database contains 501 characters. The results of each method are given in the following table:

	% of properly segmented characters
Niblack	73,85
Sauvola	71,26
TMMS	**74,85**
Ultimate Opening	48,10

Our method gives the best results. Thresholding with Sauvola criterion is far less efficient on average. It fails frequently on text correctly handled with Nilback criterion or our method but, in some situations, it gives the best quality segmentation. The overall poor result is explained by the high difficulty level of the environment. The ultimate opening surprisingly gives bad results. This may come from the fact that images are taken by sensors mounted on a moving car: images may have a motion blur, which makes the ultimate opening fail. We then cancel it from the comparison.
The other aspect of our comparison is speed. We evaluate all methods on the set of images and compute mean times. Times are given in seconds for 1920x1080 image size and according to the size of the mask of every method:

Mask size	3x3	5x5	7x7	9x9	11x11
Niblack	0,16	0,22	0,33	0,47	0,64
Sauvola	0,16	0,23	0,33	0,47	0,64
TMMS	**0,11**	**0,18**	**0,27**	**0,44**	**0,55**

All implementations are performed according to the definition without any optimization. Our method always gets the best execution times (Notice that Shafait et al. (Shafait et al., 2008) have recently offered a faster way to compute Sauvola criterion).

The speed of the algorithm is important but the output is also a major aspect as execution time of a complete scheme usually depends on the number of regions provided by segmentation steps. On our database, on average, binarization

Figure 8: Examples of text and non text samples in learning database.

with Niblack criterion generates 65177 regions, binarization with Sauvola criterion generates 43075 regions, our method generates 28992 regions. Reducing the number of regions in the output may save time when we process these regions. The possibility, in our method, to set up the lowest allowed contrast prevents from having over segmented regions. Moreover, many of these regions, noticed as homogeneous, can be associated with other neighbour regions (end of section 3). This simple process may lead to a decrease in the number of regions. This low number of regions may increase the localisation precision as it can decrease false positives. It is another proof that the segmentation provided by our method is more relevant.

Letter Classification To perform training and testing we have constituted (Fig. 8):

- a training data base composed of 32400 examples with 16200 characters from various sources (letters at different scales/points of view...) and 16200 other regions extracted from various urban images and,
- a testing base with 3600 examples.

Notice that all training are performed by tools provided by (Joachims, n.d.).
Different configurations of classifiers have been tested to get the highest classification accuracy. With the configuration we have chosen (Figure 7), the svm classifier trained with pseudo Zernike moments gives 75.89% of accuracy, the svm classifier trained with our polar descriptors gives $81,50\%$ of accuracy and last svm classifier trained with Fourier descriptors gives $83,14\%$ of accuracy. This proves that our descriptor is well defined as its accuracy is at the same level of accuracy as Fourier descriptors and pseudo Zernike moments.
To make the final decision we choose a *late fusion* architecture. Different tests are performed: from a simple vote of the three previous classifiers to the use of another classifier. The best result has been reached by the use of a SVM classifier which gets, $87,83\%$ of accuracy with the confusion matrix :

%	Letter	Background
Letter	91,56	8,44
Background	15,89	84,11

The unbalanced result is interesting for us, as the most important for us is not to lose a character.

Figure 9: The system localizes correctly text in the image (even with rotated text) but it detects aligned windows as text.

Figure 10: Text is correctly localized, but the classification step fails on the end of the word *courant* in red and zebra crossing sign is seen as text.

We also test different combinations of classifiers and descriptors. When we try early fusion architecture, we give all descriptors to a unique svm classifier ; the result does not even reach 74% of accuracy. On the contrary, if we add a collection of simple geometric descriptors (compacity, surface, concavity...) to the svm classifier that must take the final decision in our architecture, the overall accuracy reaches $88,83\%$. These measures seem to help the classifier to select which classifiers are the most reliable depending on the situation.
The overall accuracy seems to be a bit low but the variability of text in our context is so huge that the real performance of the system is not so bad.

8 TEXT LOCALIZATION IN CITY SCENES

Let us see the application of the complete scheme. We took an initial image (Figure 12). The application of our algorithm of segmentation gives the result in figure 13. All regions with a reasonable size are kept, others are dismissed (Figure 14). The classifier selects text regions among remaining regions (Figure 15). Text regions are grouped to create words and sentences (Figure 16).
The system is efficient: instead of a variation of orientation, police and lighting condition, the system handles majority of text (Figure 9, 10 et 11). But it also generates many false positives: especially aligned windows (Figure 9 top right and Figure 11). Other results can be seen in figures 9 and 10. The system must then be improved to reduce false positives.

Figure 11: Various texts are correctly handled but periodical features are also interpreted as text.

9 CONCLUSION

We have presented a text localization process defined to be efficient in the difficult context of the urban environment. We use a combination of an efficient segmentation process based on morphological operator and a configuration of svm classifiers with various descriptors to determine regions that are text or not. The system is competitive but generates many false positives. We are currently working to enhance this system (and reducing false positives) by improving the last two steps: we keep on testing various configurations of classifiers (and selecting kernels of svm classifiers) to increase the accuracy of the classifier and we are especially working on a variable selection algorithm. We are also working on the grouping step of neighbour text regions and its correction to send properly extracted text to O.C.R.

ACKNOWLEDGEMENTS

We are grateful for support from the French Research National Agency (A.N.R.)

REFERENCES

Arth, C., Limberger, F. and Bischof, H., 2007. Real-time license plate recognition on an embedded DSP-platform. IEEE International Conference on Computer Vision and Pattern Recognition (CVPR '07) pp. 1–8.

Beucher, S., 2007. Numerical residues. Image Vision Comput. 25(4), pp. 405–415.

Chen, X. and Yuille, A. L., 2004. Detecting and reading text in natural scenes. Computer Vision and Pattern Recognition, IEEE Computer Society Conference on 2, pp. 366–373.

Cortes, C. and Vapnik, V., 1995. Support-vector networks. Machine Learning 20(3), pp. 273–297.

ImagEval, 2006. www.imageval.org.

Institut Géographique National, n.d. www.ign.fr.

iTowns ANR project, 2008. www.itowns.fr.

Jian Liang, David Doermann and Huiping Li, 2005. Camera-Based Analysis of Text and Documents: A Survey. International Journal on Document Analysis and Recognition 7(2+3), pp. 83 – 104.

Joachims, T., n.d. svm. http://svmlight.joachims.org/.

Jung, K., Kim, K. and Jain, A., 2004. Text information extraction in images and video: a survey. Pattern Recognition 37(5), pp. 977–997.

Kavallieratou, E., Balcan, D., Popa, M. and Fakotakis, N., 2001. Handwritten text localization in skewed documents. In: International Conference on Image Processing, pp. I: 1102–1105.

Mancas-Thillou, C., 2006. Natural Scene Text Understanding. PhD thesis, TCTS Lab of the Facult Polytechnique de Mons, Belgium.

Niblack, W., 1986. An Introduction to Image Processing. Prentice-Hall, Englewood Cliffs, NJ.

Palumbo, P. W., Srihari, S. N., Soh, J., Sridhar, R. and Demjanenko, V., 1992. Postal address block location in real time. Computer 25(7), pp. 34–42.

Retornaz, T. and Marcotegui, B., 2007. Scene text localization based on the ultimate opening. International Symposium on Mathematical Morphology 1, pp. 177–188.

Sauvola, J. J., Seppänen, T., Haapakoski, S. and Pietikäinen, M., 1997. Adaptive document binarization. In: ICDAR '97: Proceedings of the 4th International Conference on Document Analysis and Recognition, IEEE Computer Society, Washington, DC, USA, pp. 147–152.

Serra, J., 1989. Toggle mappings. From pixels to features pp. 61–72. J.C. Simon (ed.), North-Holland, Elsevier.

Shafait, F., Keysers, D. and Breuel, T. M., 2008. Efficient implementation of local adaptive thresholding techniques using integral images. Document Recognition and Retrieval XV.

Szumilas, L., 2008. Scale and Rotation Invariant Shape Matching. PhD thesis, Technische universitt wien fakultt fr informatik.

Wahl, F., Wong, K. and Casey, R., 1982. Block segmentation and text extraction in mixed text/image documents. Computer Graphics and Image Processing 20(4), pp. 375–390.

Wolf, C., michel Jolion, J. and Chassaing, F., 2002. Text localization, enhancement and binarization in multimedia documents. In: In Proceedings of the International Conference on Pattern Recognition (ICPR) 2002, pp. 1037–1040.

Figure 12: The initial image used for the test. This image is provided by the french ign (Institut Géographique National, n.d.).

Figure 13: The image segmented by our algorithm TMMS.

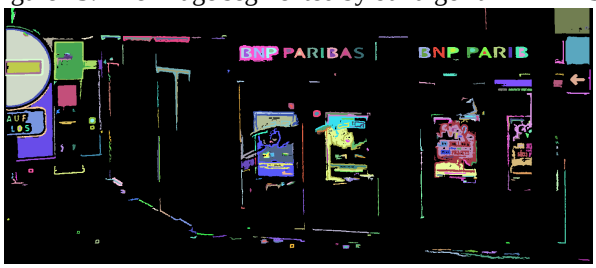


Figure 14: All big regions are removed. Only the regions of reasonable size are kept.

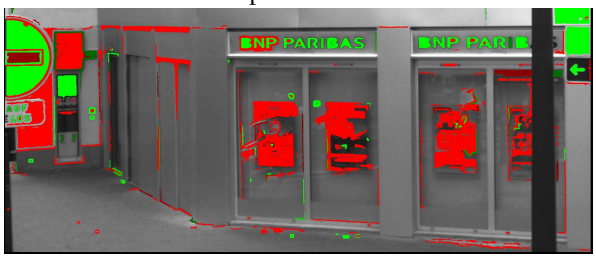


Figure 15: Remaining regions are classified by our system. Text region (in green) are kept, non text region (in red) are removed.

Figure 16: Isolated text regions are removed and remaining regions are grouped.

CIRCULAR ROAD SIGN EXTRACTION FROM STREET LEVEL IMAGES USING COLOUR, SHAPE AND TEXTURE DATABASE MAPS

A. Arlicot *, B. Soheilian and N. Paparoditis
Institut Géographique National, Laboratoire MATIS, 73, avenue de Paris, 94165 Saint-Mandé cedex, France
aurore.arlicot@polytech.univ-nantes.fr, bahman.soheilian@ign.fr, nicolas.paparoditis@ign.fr
http://recherche.ign.fr/labos/matis/

KEY WORDS: mobile mapping system, road sign recognition, color detection, ellipse detection, pattern matching

ABSTRACT

Detection and recognition of road signs can constitute useful tools in driving assistance and autonomous navigation systems. We aim at generating a road sign database that can be used for both georeferencing in autonomous vehicle navigation systems and also in high scale 3D city modelling. This paper proposes a robust algorithm that can detect road signs shape and recognizes their types.

1 INTRODUCTION

Road signs are very important features for providing rules of navigation. Indeed, they are key landmarks when navigating on the roads. Their visual properties are very strong because they have been designed to be remarkable and unmissible objects. Road signs are thus key objects to enrich road model databases to generate roadbooks, shortest paths, etc. The automatic detection and recognition of road signs from images (together with objects such as road marks) is thus a key topic and issue for road model updating but also for tomorrow's applications of these databases, i.e. driving assistance, and accurate localisation functions for autonomous navigation. Most of the previous work in image based road sign extraction deal with three following issues:

- **Color detection :** road signs are often red or blue with some black and white. Many authors used this property to detect them. Often, color base rules are defined in a color space and used for segmentation. (de la Escalera, 1997) use RGB color space and work with relations between the red , green and blue. Other authors works with color spaces that are less sensitive to lighting changes. Although the HSI (Hue, Saturation, Intensity) space is the most common (Piccioli et al., 1996). More complicated color space such as LCH (Lightness, Chroma, Hue) (Shaposhnikov et al., 2002) and CIELAB (Reina et al., 2006) are also used.
- **Shape detection:** road signs forms are often rectangular, triangular or circular. In order to strengthen the detection, some authors propose to detect these geometric forms within ROIs[1] provided by color detection. (Ishizuka and Hirai, 2004) present an algorithm for circular road sign detection. (Habib and Jha, 2007) propose an algorithm for road sign forms detection by line fitting. An interesting measure of ellipticity, rectangularity, and triangularity is proposed by (Rosin, 2003).
- **Type recognition:** It consists in recognising road sign type using its pictorial information. It is often performed by comparing the inside texture of a detected road sign with the textures in a database. For this purpose different kind of algorithms are used in the state of the art. (Priese et al., 1995) propose an algorithm that is based on neural networks. SIFT descriptors are used by (Aly and Alaa, 2004). (de la Escalera et al., 2004) used intensity correlation score as a measure of similarity to compare the detected road sign with a set of standard signs.

2 OUR STRATEGY

We propose an algorithm consisting in three main steps. Diagram of Figure 1 shows the pipeline of our algorithm. First step uses color properties of signs and perform a predetection (Section 3). It provides a set of ROIs in image space. Then, an ellipse detection algorithm is applied to detect circular shape signs within the ROIs (Section 4). The detected shapes are considered as road sign hypotheses. Final step consists in validation or rejection of hypotheses. This is performed by matching detected hypotheses with a set of standard circular signs of the same color (Section 5). Results and evaluations are presented in Section 6.

3 COLOR DETECTION

A large number of road signs are blue or red. It can simplify their detection by looking for red and blue pixels. However their RGB values depend on illumination conditions. We use HSV (Hue, Saturation, Value, see Equation 1) color space because it is robust against variable conditions of luminosity. In order to choose the adapted threshold of saturation and hue, we learn these parameters from a set of road sign sample in different illumination conditions. Figure 2(a) shows our running example image and result of blue color detection is shown in Figure 2(b). In order to provide ROIs, connected pixels are labeled (see Figure 2(c)). Each label defines a window in image space. The following form detection and validation steps are performed within these windows.

*A. Arlicot is currently at Polytech'Nantes, IRCCyN lab France.
[1]Region of Interest

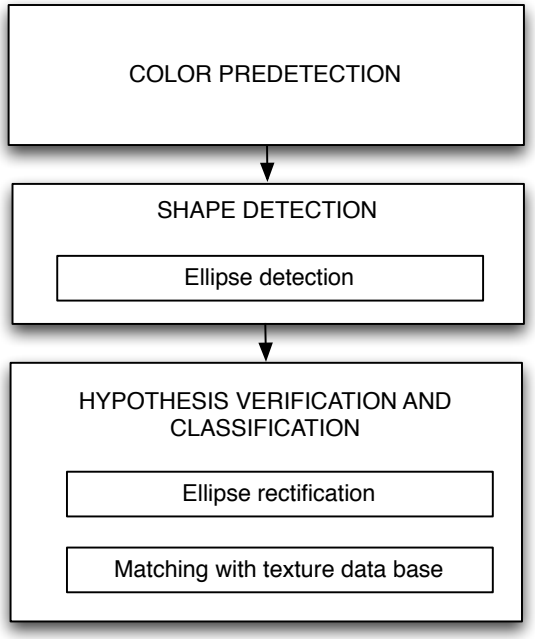


Figure 1: Our 3 steps strategy.

$$H = \begin{cases} (0 + \frac{G-B}{MAX-MIN}) \times 60 & \text{if } R = MAX, \\ (2 + \frac{B-R}{MAX-MIN}) \times 60 & \text{if } G = MAX, \\ (4 + \frac{R-G}{MAX-MIN}) \times 60 & \text{if } B = MAX, \end{cases}$$

$$S = \frac{MAX - MIN}{MAX},$$

$$V = MAX,$$

where:

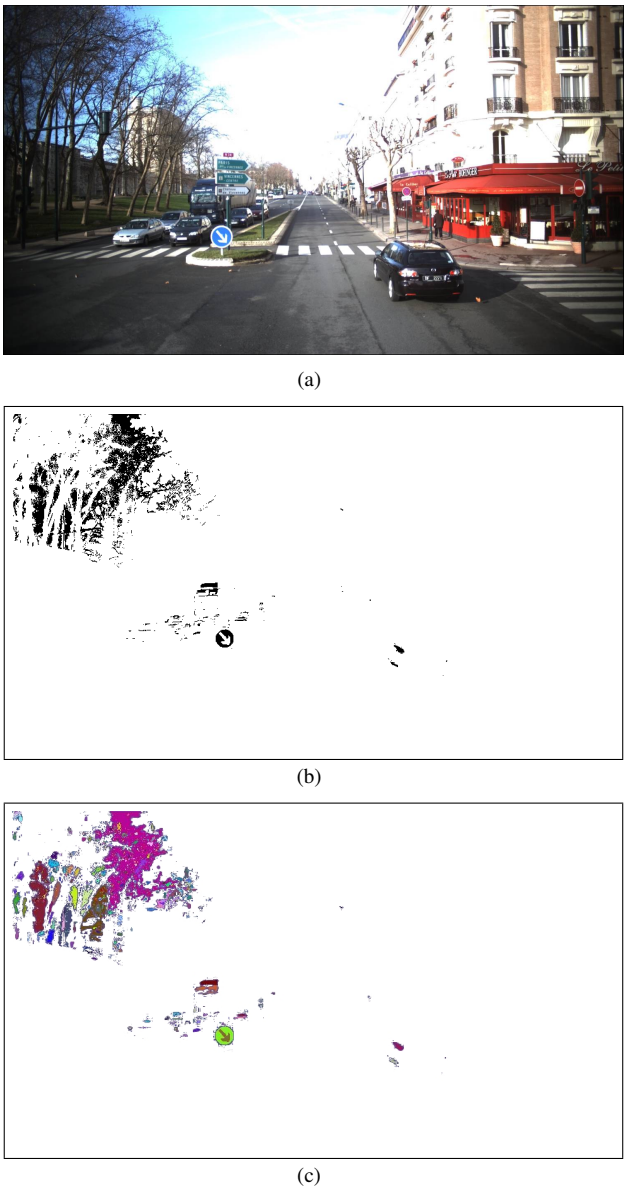
$$MAX = max(R, G, B)$$

$$MIN = min(R, G, B) \quad (1)$$

4 CIRCULAR SIGN DETECTION

The shape detection have to detect all the types of road signs (the rectangular, triangular and circular road signs). In this first version of work we choose to focus on the circular road signs because they are the most common. Theorically, a circle appears as an ellipse in perspective images. The quantity of perspective deformation depends on the angle between image and the circle plane. Often, road signs belong to a traffic lane and supposed to provide information to drivers in the same lane. In this case perspective deformation is negligible. This is the reason why most of the Driver Assistance Systems (ADAS) ignore perspective deformation.

We aim at extracting all visible road signs within an image what ever their orientation is. This is interesting in both database generation and the use of road signs as visual landmarks for positioning purposes. Thus, an ellipse detection algorithm is investigated (Section 4.1).



(a)

(b)

(c)

Figure 2: Color detection results. a) our running example RGB image, b) blue color mask, c) labeling independent connected pixels.

4.1 Ellipse Detection

Input of this step is a set of image windows provided by the color detection step. We use edge points for ellipse detection. In each image window, edges are extracted using Canny-Deriche edge detector (Deriche, 1987).

An ellipse is defined with five parameters (2 for the center, 2 for the axes length and one for orientation). Equation 2 express equation of ellipse. In this Equation p and q stand for ellipse center. Orientation and axes length depend on a, b and c.

$$a(x-p)^2 + 2b(x-p)(y-q) + c(y-q)^2 = 1 \quad (2)$$

This equation is not linear. We make use the Pascal's the-

orem to find the center (p, q) of the ellipse using only 3 points by estimation of tangents at each point. It allows a linear estimation of ellipse using only 3 points.

4.1.1 Ellipse from three points Given 3 points P_1, P_2, P_3 on an ellipse (see Figure 3) the center is computed as follows:

- Tangents at these 3 points (t_1, t_2, t_3) are found.
- Intersections of t_1 with t_2 (I_1) and t_2 with t_3 (I_2) are computed.
- Midpoints of the segments $[P_1P_2]$ and $[P_2P_3]$ (M_1 and M_2) are found.
- The intersection of the segments $[I_1M_1]$ and $[I_2M_2]$ gives the ellipse center (C).

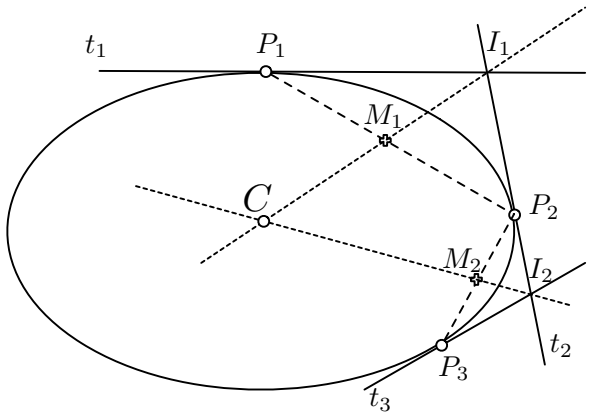


Figure 3: Use of Pascal's theorem for estimating ellipse center with 3 points.

When the center coordinates (p, q) are obtained the coordinate system is shifted such as (p, q) become origin. Then, the Equation 3 can be applied to estimate the ellipse equation using 3 points.

$$ax^2 + 2bxy + cy^2 = 1 \tag{3}$$

4.1.2 Ellipse estimation with RANSAC In the previous section the ellipse estimation method was explained when we have three points on the ellipse. The problem is to obtain three points belonging to the ellipse within the noise (see Figure 4(a)). We used a RANSAC algorithm (Fischler and Bolles, 1981). It is composed of six steps:

1. Pick randomly three points within the edges points.
2. Estimate the ellipse parameters (see Section 4.1.1).
3. Search how many edge points fit on the ellipse model (number of support points).
4. If the number of support point is sufficiently great, we accept the model and exit the loop with success. We assume that the number of support point is sufficient when it is higher than a percentage of the estimated theoretical ellipse circumference.
5. Repeat the steps 1 to 4, n times.
6. If we arrive to this step, we declare a failure and there is no ellipse found.

Suppose that the density ratio of inlier is 50% and the probability that the algorithm exit without finding a good fit is chosen 5%, then, the number of needed iterations (n) is 25.

In ellipse estimation, in order to compute the needed tangent on each edge point, a line is fitted to its neighbours on the linked edges. A neighborhood of 2 pixels is chosen. Due to discretisation, it does not provide a good tangent estimation when using pixel accuracy. This problem is shown in Figures 4(b) and 4(c). It causes more frequent failure and less accurate result. In order to cope with this problem, the edge points are delocalised to provide a subpixel accuracy using the method developed in (Devernay, 1995).

Figure 5 shows an example of result obtained by this algorithm.

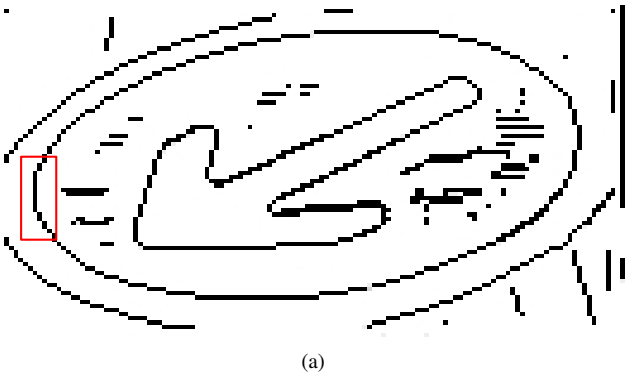
5 HYPOTHESIS VERIFICATION AND TEXTURE PATTERN RECOGNITION

5.1 Ellipse Rectification

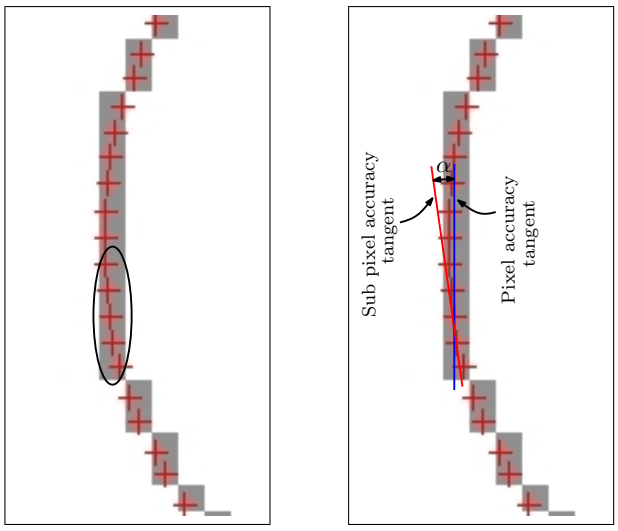
Validation and recognition of road sign is performed by comparing the detected circular road sign with a set of reference ones (See Figure 7) . The inside texture of sign is used to measure the its similarity with all reference signs. Correlation coefficient seems to be particularly interesting for this purpose. However the detected signs are deformed to ellipse while the reference ones are circular. It make the correlation process difficult. In order to resolve the problem, we propose to rectify the texture of the detected sign to match the geometry of reference ones. The needed transformation must transform an ellipse to a circle of a given radius. This is performed using an 8 parameters projective transformation. We suppose that the images are approximately horizontal or the orientation of the images are known so the transformation is unique. Figure 6 shows some examples of resampled road signs.

5.2 Matching with texture DB

After rectification, in order to match only the pixel inside the road sign, we generate a circular mask and we apply the ZMNCC (Zero Mean Normalized Cross Correlation) function to compute the similarity of detected and reference object (See Equation 4).

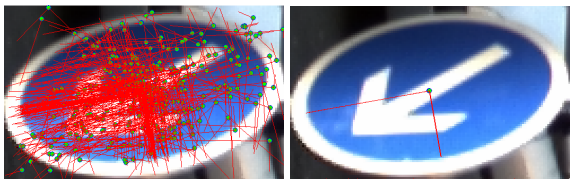


(a)



(b) (c)

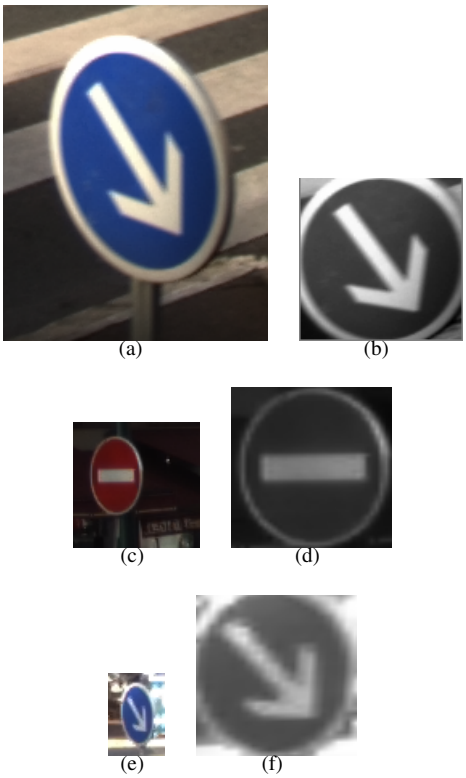
Figure 4: Edge extraction: red crosses represent subpixel accuracy edge position. : a) extracted edges, b) a zoom on edges of (a), 5 points are chosen for tangent estimation, c) Difference between pixel accuracy tangent and sub pixel one.



(a) (b)

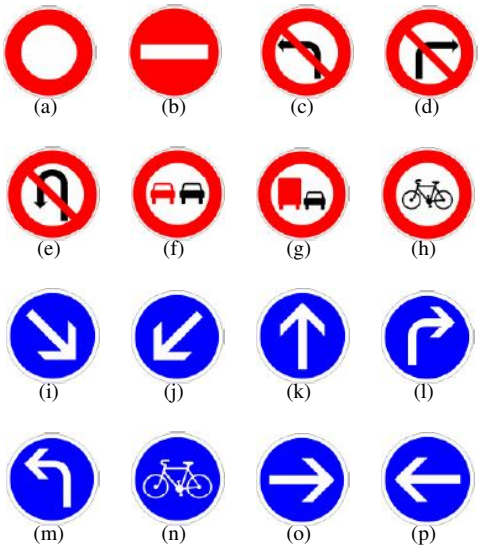
Figure 5: (a) Example of all the centers and axes explored by RANSAC algorithm (b) the estimated solution.

Figure 8 shows some result of correlation. We match detected red signs only with the red reference signs and blue ones with blue references. However in Figure 8, correlation coefficient with all signs are shown to demonstrate the discrimination power of correlation function. In most of the cases, the maximum of correlation coefficient corresponds to the good sign. We accept the maximum of correlation if it is higher than 60%. Hypotheses with lower correlation coefficients are rejected. This threshold is chosen relatively low. The reason is that the texture of signs in images suffer from both radiometric calibration problem and illumination changes within one sign. Better radiometric calibration can partially reduce this effect. So higher correlation coefficient thresholds can be set in the algorithm and improve the reliability of recognition.



(a) (b) (c) (d) (e) (f)

Figure 6: (a), (c) and (e) are the original image windows and (b), (d) and (f) are respectively their resampled images.



(a) (b) (c) (d) (e) (f) (g) (h) (i) (j) (k) (l) (m) (n) (o) (p)

Figure 7: Circular road signs reference database.

$$Score_{corr}(A,B) = \frac{\sum_{x=1}^{n}\sum_{y=1}^{m}[A(x,y)-\overline{A}][B(x,y)-\overline{B}]}{\sqrt{\sum_{x=1}^{n}\sum_{y=1}^{m}[A(x,y)-\overline{A}]^2 \sum_{x=1}^{n}\sum_{y=1}^{m}[B(x,y)-\overline{B}]^2}} \quad (4)$$

6 RESULTS AND PERFORMANCE EVALUATION

The proposed algorithm is evaluated on a set of 1370 images acquired in dense urban area with real traffic conditions. Figures 9-13 show some obtained results. In each image the number of correct detection, false detection, and true road signs are counted manually. We assume that if a road sign is smaller than 10 pixels, we can not detect it.

We observed that there is 67% of good detection and 33% of road signs are not detected. This is due to our camera radiometric calibration problems that causes color detection failure. As color detection is at the beginning of our pipeline the shape detection and recognition processes are not performed on the lost road signs.

The shape detection and recognition steps works well. We mean that, in most of the cases they reject correctly the false hypotheses and in the case of validation the type of road signs are correctly distinguished. However, there is 5% of false detection. They are in most of the cases due to the red lights behind the cars or the tricolor lights that are very similar to wrong-way (see Figure 7(b)) traffic sign (see Figure 9) .

7 CONCLUSION AND TRENDS

In this paper we proposed a pipeline for road sign detection in RGB image. Thanks to ellipse detection and rectification processes, the algorithm is not sensitive to road sign orientation. The matching step provides a reliable recognition of road sign type.
Evaluations revealed that, the detection rate is about 70%. This is always due to failure in color detection step. Better radiometric calibration of the camera and test of other color spaces are the work in progress for improving color detection. In contrast to color detection step our shape detection and recognitions steps provide satisfactory and reliable results.
The proposed algorithm can be easily extended to handle the rectangular and triangular road signs. For this purpose, it is enough to adapt the shape detection step and both other steps remain unchanged.
In Figure 13 we can see a particular case which represent two small road signs on a bigger road sign. These cases can be handled using a stereo system allowing 3D position and size estimation.
In real time applications such as driver assistance systems, it is often interesting to track objects in video sequences. Actually, our algorithm does not work in real time and can not be applied on video sequences. The edge detection is the most time consuming step. In order to reduce the processing time, other edge detectors such as Sobel or Prewitt filters can be applied and evaluated. The search area can also be limited to remove the sky and so speed up the global processing time.

REFERENCES

Aly, F. and Alaa, A., 2004. Detection, categorization and recognition of road signs for autonomous navigation. In: Proceeding of Advanced Concepts for Intelligent Vision System, Brussels, Belgium.

de la Escalera, A., Armingol, J., Pastor, J. and Rodriguez, F., 2004. Visual sign information extraction and identification by deformable models for intelligent vehicles. IEEE Transactions on Intelligent Transportation Systems 5(2), pp. 57–68.

de la Escalera, A. Moreno, L. S. M. A. J., 1997. Road traffic sign detection and classification. IEEE Transactions on Industrial Electronics.

Deriche, R., 1987. Using canny's criteria to derive a recursively implemented optimal edge detector. The International Journal of Computer Vision 1(2), pp. 167–187.

Devernay, F., 1995. A non-maxima suppression method for edge detection with sub-pixel accuracy. Technical Report RR-2724, INRIA.

Fischler, M. A. and Bolles, R. C., 1981. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. Communications of the ACM 24(6), pp. 381–395.

Habib, A. and Jha, M., 2007. Hypothesis generation of instances of road signs in color imagery captured by mobile mapping systems. International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences 36 part 5/C55) pp. 159–165.

Ishizuka, Y. and Hirai, Y., 2004. Segmentation of road sign symbols using opponent-color filters. In: ITSWC, Nagoya, Japon.

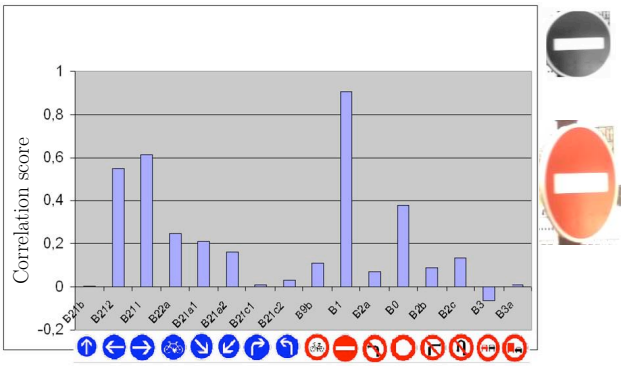
Piccioli, G., Micheli, E. D., Parodi, P. and Campani, M., 1996. Robust method for road sign detection and recognition. Image Vision Comput. 14(3), pp. 209–223.

Priese, L., Lakmann, R. and Rehrmann, V., 1995. Ideogramm identification in a realtime traffic sign recognition system. In: Proceeding of intelligent vehicles apos, IEEE, Nagoya, Japon.

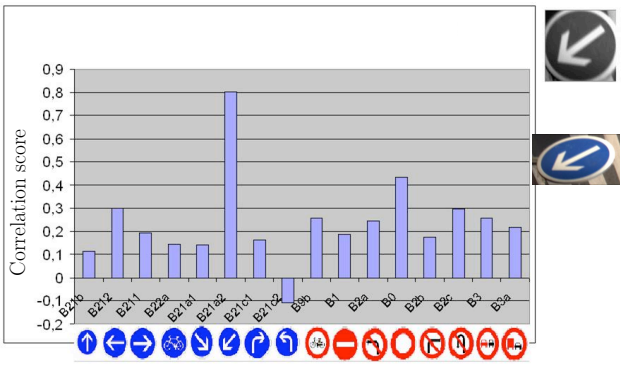
Reina, A. V., Sastre, R. J. L., Arroyo, S. L. and Jiménez, P. G., 2006. Adaptive traffic road sign panels text extraction. In: ISPRA'06: Proceedings of the 5th WSEAS International Conference on Signal Processing, Robotics and Automation, World Scientific and Engineering Academy and Society (WSEAS), Stevens Point, Wisconsin, USA, pp. 295–300.

Rosin, P. L., 2003. Measuring shape: ellipticity, rectangularity, and triangularity. Machine Vision and Applications 14(3), pp. 172–184.

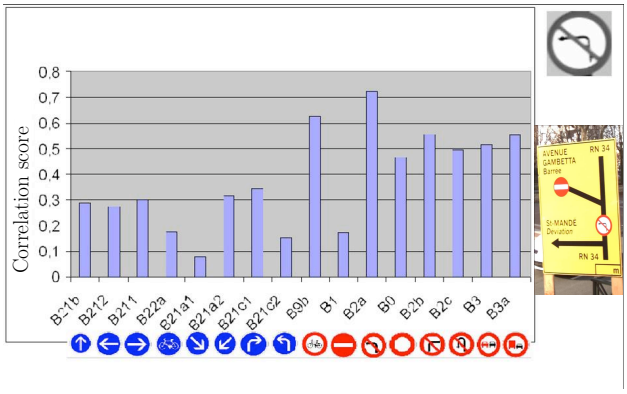
Shaposhnikov, D., Podladchikova, L., Golovan, A. and Shevtsova, N., 2002. Road sign recognition by single. positioning of space-variant sensor window. In: Proc. 15th International Conference on Vision Interface, Calgary, Canada, pp. 213–217.



(a)



(b)



(c)

Figure 8: Correlation score of the hypotheses with road sign DB.

Figure 9: A false detection example. Red light of tri color light is detected as wrong way traffic sign.

Figure 10: Detection of road signs.

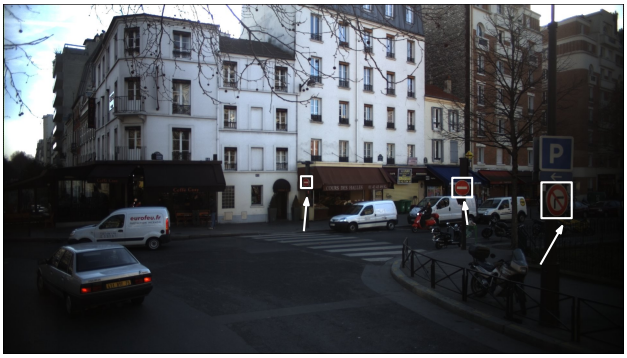


Figure 11: Detection of red road signs.

Figure 12: Detection of blue road signs.

Figure 13: Detection of particulars road signs.

IMPROVING IMAGE SEGMENTATION USING MULTIPLE VIEW ANALYSIS

Martin Drauschke, Ribana Roscher, Thomas Läbe, Wolfgang Förstner
Department of Photogrammetry, Institute of Geodesy and Geoinformation, University of Bonn
martin.drauschke@uni-bonn.de, rroscher@uni-bonn.de, laebe@ipb.uni-bonn.de, wf@ipb.uni-bonn.de

KEY WORDS: Image Segmentation, Aerial Image, Urban Scene, Reconstruction, Building Detection

ABSTRACT

In our contribution, we improve image segmentation by integrating depth information from multi-view analysis. We assume the object surface in each region can be represented by a low order polynomial, and estimate the best fitting parameters of a plane using those points of the point cloud, which are mapped to the specific region. We can merge adjacent image regions, which cannot be distinguished geometrically. We demonstrate the approach for finding spatially planar regions on aerial images. Furthermore, we discuss the possibilities of extending of our approach towards segmenting terrestrial facade images.

1 INTRODUCTION

The interpretation of images showing building scenes is a challenging task, due to the complexity of the scenes and the great variety of building structures. As far as human perception is understood today, humans can easily group visible patterns and use their shape to recognize objects, cf. (Hoffman and Richards, 1984) and (Treisman, 1986). Segmentation, understood as image partitioning often is the first step towards finding basic image patterns. Early image segmentation techniques are discussed in (Pal and Pal, 1993). Since then, many other algorithms have been proposed within the image analysis community: The data-driven approaches often define grouping criteria based on the color contrast between the regions or on textural information. Model-driven approaches often work well only on simple scenes e. g. simple building structures with a flat or gabled roof. However, they are limited when analyzing more complex scenes.

Since we are interested in identifying entities of more than two classes as e.g. buildings, roads and vegetation objects, we cannot perform a image division into fore- and background as summarized in (Sahoo et al., 1988). Our segmentation scheme partitions the image into several regions.

It is very difficult to divide an image into regions if some regions are recognizable by a homogenous color, others have a significant texture, and others are separable based on the saturation or the intensity, e. g. (Fischer and Buhmann, 2003) and (Martin et al., 2004). However, often such boundaries are not consistent with geometric boundaries. According to (Binford, 1981), there are seven classes of boundaries depending on illumination, geometry and reflectivity. Therefore, geometric information should be integrated into the segmentation procedure.

Our approach is motivated by the interpretation of building images, aerial and terrestrial, where many surface patches can be represented by low order polynomials. We assume a multi-view setup with one reference image and its intensity based segmentation, which is then improved by exploiting the 3D-information from the depth image derived from all images. Using the determined orientation data, we are able to map each 3D point to an unique region. Assuming, object surfaces are planar in each region, we can estimate a plane through the selected points. The adjacent regions are merged together if they have similar planes. Finally, we obtain an image partition with regions representing dominant object surfaces as building parts or ground. We are convinced that the derived regions are much better for an object-based classification than the regions of the initial segmentation, because many regions have simple, characteristic shapes.

The paper is structured as followed. In sec. 2 we discuss recent approaches of combining images and point cloud information, mostly with the focus on building reconstruction. Then in sec. 3 we briefly sketch our approach for deriving a dense point cloud from three images. So far, our approach is semi-automatic due to the setting of the point cloud's scale, but we discuss the possibility of automatization for all its steps. In sec. 4 we present how we estimate the most dominant plane in the dense point cloud restricted on those points, which are mapped to pixels of the same region. The merging strategy is presented in sec. 5. Here we only study the segmentation of aerial imagery and present our results in sec. 6. Adaptations for segmenting facade images are discussed in each step separately. We summarize our contribution in the final section.

2 COMBINING POINT CLOUDS AND IMAGES

The fusion of imagery with LIDAR data has successfully be done in the field of building reconstruction. In (Rottensteiner and Jansa, 2002) regions of interests for building extraction are detected in the set of laser points, and planar surfaces are estimated in each region. Then the color information of the aerial image is used to merge adjacent coplanar point cloud parts. Contrarily, in (Khoshelham, 2005) regions are extracted from image data, and the spatial arrangement of corresponding points of a LIDAR point cloud is used as a property for merging adjacent regions. In (Sohn, 2004) multispectral imagery is used to classify vegetation in the LIDAR point cloud using a vegetation index. The advantage of using LIDAR data is to work with high-precision positioned points and a very limited portion of outliers. The disadvantage is its expensive acquisition, especially for aerial scenes. Hence, we prefer to derive a point cloud from multiple image views of an object.

Within the last years, the matching of multiple views of an object enabled the reconstruction of 3D object points with high accuracy and high density. Previous approaches as (Kanade and Okutomi, 1994) are based on a low-level preprocessing of the image to extract points of interest. Then, the correspondences of such points are used to estimate the 3D position of the object points. In many applications, Förstner-features (Förstner and Gülch, 1987) or SIFT-features (Lowe, 2004) are used, but the derived point clouds are either sparse or have been extracted from many images or video, e. g. (Mayer and Reznik, 2005) and (Gallup et al., 2007). In (Tuytelaars and Van Gool, 2000), the correspondences are determined over local affinely invariant regions, which were extracted from local extrema in intensity images. This procedure is liable to make matching mistakes if the image noise is relatively high.

Dense point clouds from only a few images are obtained by adjusting the correspondence between pixels by correlation based on (semi-) global methods, e. g. (Hirschmüller, 2005). Assuming the observed objects have a smooth surface, the accuracy of the obtained point clouds gets increased by including information on the relations between the pixels by a Markov random field, e. g. (Yang et al., 2009), or from image segmentation, e. g. (Tao and Sawhney, 2000).

In our approach, we take up the idea of (Khoshelham, 2005) to improve an initial image segmentation using additional 3D information. From multi-view analysis, we derive a point cloud, which is used for deriving additional features for the segmented image regions. We focus on building scenes, whose objects mostly consist of planar surfaces. So, it is reasonable to look for dominant planes in the point cloud, where the search is guided by the image segmentation.

For us, it is important to realize an approach, which has the potential to get automatized since there are many applications with thousands of images. There is a need for a completely automatic procedure if additional features are derived from a reconstructed point cloud to improve the segmentation or interpretation of the images. Our input are only two or more images from the object, which were taken by a calibrated camera. An example is shown in fig. 1.

3 RECONSTRUCTION OF THE 3D SCENE

In this section, we describe the generation of the point cloud $\boldsymbol{C}$ from the given images. For this generation, there are two conditions, which should be fulfilled: (a) the observed objects should be textured sufficiently and (b) the views must overlap, otherwise we have problems to determine the relative orientation between the images. So far, the implemented algorithms need some human interaction for setting the point cloud scale and the disparity range parameters, but under certain conditions, the whole approach could get designed to perform completely automatically.

We describe the procedure with two or three given images $\boldsymbol{I}_1$, $\boldsymbol{I}_2$ and $\boldsymbol{I}_3$. Two views are necessary to reconstruct the

Figure 1: Three aerial views of a building scene consisting of a flat roofed part and a gable roofed part. The initial segmentation of the upper view is shown on its right side. The ground consists of several weirdly shaped regions, and the flat roof is also not well segmented.

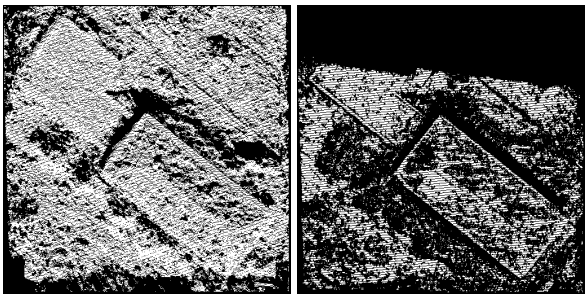


Figure 2: Reconstructed 3D-points are projected back into 2D-image (white). Left: all pairs of matches are shown. The point cloud is very dense with approximately 75% of pixels having a 3D point, but these points are very imprecise. Right: only matches in all three images are shown. The point cloud is still dense with approximately 30% of pixels having a 3D point with higher precision.

observed 3D data, but if the matching is performed over three images, the point cloud is still dense, see fig. 2, and it contains more reliable points, thus less outliers. The reconstruction process can get improved if even more images are considered. If all used images were taken by a calibrated camera, we are able to reconstruct the 3D scene by performing the following steps.

In the first step we determine the relative orientations between the given images. Of course, it can be skipped if the projection matrices have been estimated during image acquisition. Otherwise, due to the calibration of the camera we eliminate automatically the non-linear distortions using the approach of (Abraham and Hau, 1997). The matching of extracted key-points using the approach of (Lowe, 2004) leads to the determination of the relative orientations of all images, i. e. their projection matrices $\boldsymbol{P}_n$, cf. (Läbe and Förstner, 2006). The success of the relative orientation can

be evaluated according to the statistics of the performed bundle adjustment. This step is usually robust enough for typical building scenes, because the facades are often sufficiently textured, and we do not have to deal with total occlusions. Otherwise, problems may occur due to too large mirroring facade parts.

The images are oriented relatively, not absolutely, i. e. the position of the projection centers are not correctly scaled yet. Since we cannot invert a transformation from 3D to 2D, a reasonable assumption about the scale always has to be inserted additionally. The easiest way to set the scale parameter is to measure GPS positions during the image acquisitions. Another strategy would be to measure one or more distances on the object and to identify corresponding points in the images or in the extracted point cloud later. While the first way can easily get automatized, the second one has to be done by human interaction.

From the second step on, we only use three images for a dense trinocular matching and only accept those 3D points, which were matched in all three images. Thus, we reduce many matching errors close to the image borders and avoid points corresponding to occluded surfaces. We use the semi-global matching by (Hirschmüller, 2005) in a realization by (Heinrichs et al., 2007). It is efficient, does not produce too many outliers, and returns a dense point cloud with sufficiently precise points. This approach demands that the images are arranged in a L-shaped configuration with a base image, a further one shifted approximately only horizontally and a third shifted approximately only vertically. Due to the special relation between the three given images, the search space of the matching and 3D estimation of a point is reduced to a horizontal or vertical line, respectively. So far, the two parameters of the one-dimensional search space for the depth have to be set manually before the program is started. Usually, this range lies in a small bound assuming that the flying height or the distance of a facade to the camera are restricted and do not vary much.

The semi-global matching returns a disparity map, which is used to estimate the 3D point cloud by forward intersection. There are a couple of hundred or a thousand gross errors in the determined point cloud, which can be removed under the assumption that all points lie in a certain bounding box. Besides of the remaining outliers the most extracted 3D points form spatial clusters with clearly visible ground and roof planes, cf. fig. 3. Compared with other derived point clouds from stereo aerial imagery, e. g. MatchT[1], the precision of our reconstructed points is significantly lower, but we compensate it by the higher denseness.

4 REGION-WISE PLANE ESTIMATION

In this section, we describe the estimation of the most dominant plane for each detected image region of minimum size. Thereby, any arbitrary image partitioning algorithm

[1] Automated DTM Generation Environment by inpho, cf. www.inpho.de

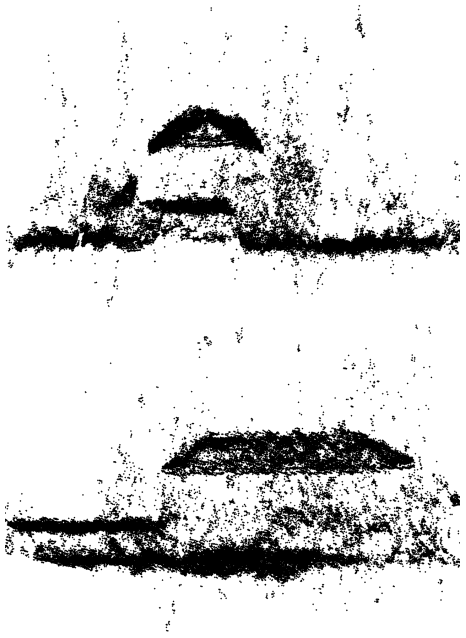


Figure 3: Side- and frontview on a point cloud, derived from scene extracts of the three aerial images from fig. 1. Besides the widely spread points on vegetation objects and some outliers, one can clearly recognize up to four major clusters showing the ground, a flat roof and a gabled roof.

can be chosen. In an earlier experiment, we made good experiences with segmenting aerial images using the watershed algorithm based on the color gradient, cf. (Drauschke et al., 2006). This segmentation approach is also applicable to facade images, cf. (Drauschke, 2009). To overcome oversegmentation at nearly all image parts, we smooth the image with a Gaussian filter with $\sigma = 2$ before determining the watershed regions. Then, oversegmented image parts are highly correlated with vegetation objects, which are not in our focus yet. Such an initial segmentation is shown in fig. 1. For further calculations, we only consider those regions R_k, which have a minimum size of 250 pixels. This parameter should depend on the image size. We have chosen a relatively high value for efficiency reasons.

In the further process, we want to estimate low order polynomial through the 3D points of each region, i. e. its most dominant plane. Therefore, we determine for each region the set of points $\{\boldsymbol{X}_j\}$ from the point cloud, which are projected into the region:

$$\boldsymbol{X}_j \mapsto R_k \;\Leftrightarrow\; \boldsymbol{x}_j = \boldsymbol{P}_n \boldsymbol{X}_j \text{ and } \boldsymbol{x}_j \in R_k. \qquad (1)$$

We assume that most dominant building surfaces and the ground are planar. Hence, we estimate the best fitting plane through the 3D points of a region. A similar procedure can be found in (Tao and Sawhney, 2000). For efficiency reason, we choose a RANSAC-based approach for our plane search, cf. (Fischler and Bolles, 1981). Therefore, we determine the parameters of the plane's normal form from three randomly chosen points $\boldsymbol{X}_{j_1}$, $\boldsymbol{X}_{j_2}$ and $\boldsymbol{X}_{j_3}$:

$$\boldsymbol{n} = (\boldsymbol{X}_{j_2} - \boldsymbol{X}_{j_1}) \times (\boldsymbol{X}_{j_3} - \boldsymbol{X}_{j_1}) \qquad (2)$$

$$d = \left\langle \frac{\boldsymbol{n}}{||\boldsymbol{n}||}, \boldsymbol{X}_{j_1} \right\rangle \qquad (3)$$

and check, how many object points support the determined plane i. e. how many points are near the plane. This depends on the choice of a threshold. Considering aerial images we allowed a maximal distance of 20 cm to the plane. If we want to guarantee with a minimum probability $p_{min} = 0.999$ finding a plane, which is constructed by 3 points and supported by at least half of the points ($\epsilon = 0.5$), we have to perform $m = 52$ trials, because

$$m = \frac{\log(1 - p_{min})}{\log(1 - (1 - \epsilon)^3)} = \frac{\log 0.001}{\log 0.875} \approx 51.7. \quad (4)$$

If no sufficiently high number of supporting points can be found within m trials, the region will no longer be analyzed. In our empirical investigation, segmented regions representing roof parts have always a most dominant plane. Such plane could not get found if e. g. the ground is not planar but forms a small hill or valley, e. g. at and around trees and shrubs. Furthermore, we accepted only those 3D points, which are visible in all three images. Therefore, occluded building parts are also not in further process.

We estimate the best fitting plane using a least-squares adjustment on those points, which support the best proposed plane during the iterations of RANSAC. The statistical reasoning[2] is taken from (Heuel, 2004), p. 145.

5 MERGING OF IMAGE REGIONS

So far, our approach can only handle with merging of regions. If the image is undersegmented in some image parts, i. e. the region covers two or more objects, a splitting criterion must be defined to separate this region parts again. We suggest to search for several dominant planes and to split the regions according to the intersections of these planes. We did not realize the splitting yet, so we only propose our merging strategy.

We determine a region adjacency graph and check for each adjacent pair of regions R_1 and R_2 if a merging of the regions can get accepted. The first test is on equality of the two corresponding estimated planes. We realized that our derived point cloud is too noisy for such statistical reasoning. Therefore, we consider a second test, where we determine the best fitting plane through the set of 3D points from both regions and then we check, if the new plane has a normal vector $\boldsymbol{n}_{12}$ which is similar to the normal vectors $\boldsymbol{n}_1$ and $\boldsymbol{n}_2$ of the two previous planes:

$$\angle(\boldsymbol{n}_{12}, \boldsymbol{n}_1) < \theta \ \wedge \ \angle(\boldsymbol{n}_{12}, \boldsymbol{n}_2) < \theta. \quad (5)$$

In our experiments, we used $\theta = 30^o$, which leads to reasonable results with respect to buildings. If one is interested in each individual roof plane, θ should not be more than 10^o. If other applications cannot depend on such a heuristically chosen parameter, we suggest to adapt this condition by a MDL-based approach, cf. (Rissanen, 1989). Then, two regions should be merged, if the encoding of data would decrease when merging.

[2]SUGR: Statistically Uncertain Geometric Reasoning, www.ipb.uni-bonn.de/projects/SUGR

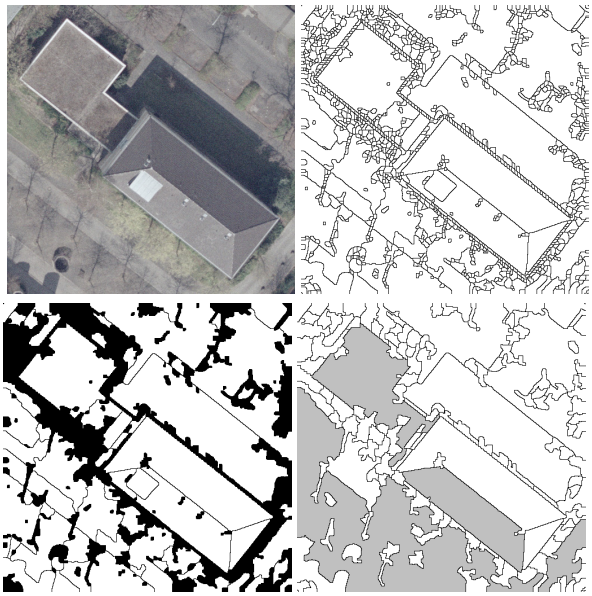


Figure 4: Steps of improving image segmentation. In the upper row, we show the reference image and its initial segmentation. In the bottom row, we show at the left all big regions from the initial partition (in white) and the final segmentation including the MDL-based and the geometry-based grouping of regions. There, the gray-shadowed regions have been merged on the basis on geometric properties.

Until this point, we did not consider small regions whose dominant planes cannot be estimated reliably. Now, we also merge them, too. Small holes can easily merge with their surrounding region, but all others may be merged according to an intensity-based criterion. We implemented a MDL-based strategy according to (Pan, 1994), where we additionally stop the merging as soon as the minimum size of a region has been reached. As alternatives, we could also use strategies for irregular pyramid structures, e. g. (Guigues et al., 2003), which is based on similarity of color intensities or (Drauschke, 2009) which is based on scale-space analysis. Resulting image segmentation is shown in fig. 4.

6 EXPERIMENTS

We have tested our segmentation scheme on 28 extracts of aerial images with known projection matrices showing urban scenes in Germany and Japan. The images from Germany were taken in early spring when many trees are in blossom, but are not covered by leaves yet. The 3D points matched at such vegetation objects are widely spread, cf. fig. 3. In most cases, the corresponding image parts are oversegmented, so that no dominant planes have to get estimated. There is almost no vegetation in the Japanese images, but the ground is often dark from shadows. As mentioned earlier, we have problems with finding precise 3D points in lawn and shadow regions, but with respect to building extraction (i. e. segmenting the major roof parts), our approach achieves satisfying results cf. fig. 5. We are convinced to get better results for matching in dark image parts, if a local enhancement is used to brighten these parts

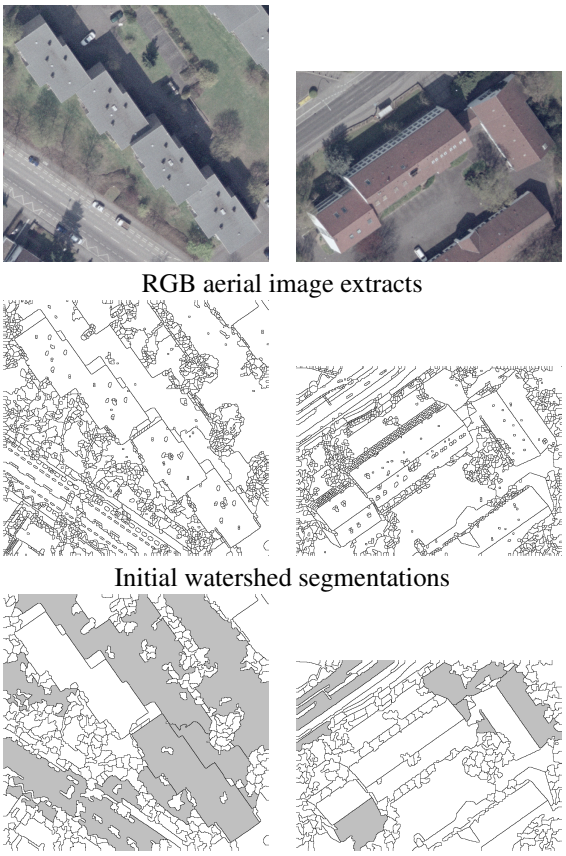


Figure 5: Results of simple building scenes. Again, the gray-shadowed regions have been merged on the basis on geometric properties.

in a preprocessing step, e. g. (Zhao and Lei, 2006). A further improvement should be acchieved, if the whole procedure is repeated, because the MDL-based merged regions are now big enough for determination of their geometric properties.

The noise of the point cloud, which we derive from the semiglobal matching does not disturb the merging of image regions. Considering aerial images, we are faced with large and often planar objects. There, our plane estimation is good enough, because we do not have to many outliers. Otherwise, the plane estimation should be done by a robust estimator. If different object parts have been segmented as one region, then the most dominant plane of the combined region often does not represent one of these object parts. This shows us, that we need to focus in the future on an algorithm for detecting multiple planes (e. g. analysis of the best five planes from RANSAC) and a splitting routine. Furthermore, there are objects as trees or dormers which violate our assumption of having one planar surface. Therefore, we consider to adapt our plane estimation towards extracting general geometric primitives as planes, cylinders, cones and spheres, cf. (Schnabel et al., 2007).

With respect to facade images, we have big trouble with our plane estimation. We ascribe this fact to two major reasons. First, the reconstruction part is challenged by homogenous facades and mirroring or light transmitting windows. Both cases lead to too many outliers. And secondly, the noise of the complete point cloud is too high to differ between planes in the object space, which are parallel, but only a view centimeters apart. Fig. 6 shows a facade image and three views on the dominant planes of given annotated objects. In this case, the supporting points may have a distance of 4 cm to the fitting plane. Dominant planes with distances of more than half of a meter are clearly separable from each other.

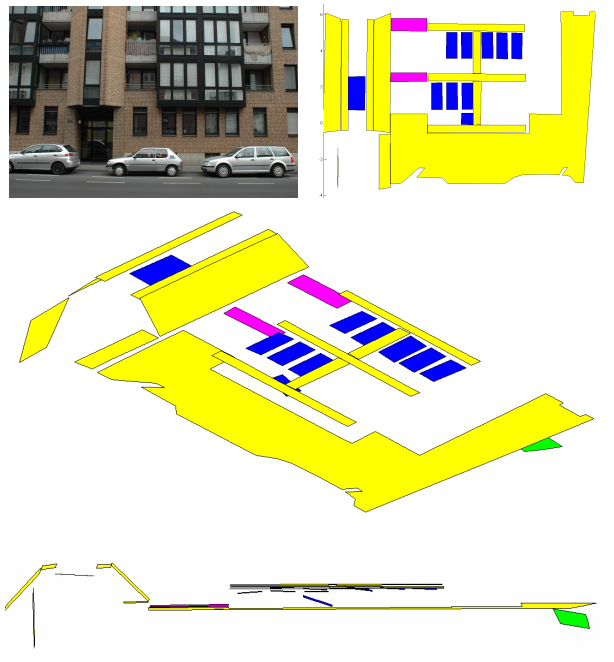


Figure 6: Facade image and different views on fitted planes for hand-labeled object parts. Wall components are drawn in yellow, windows in blue and (if opened) in green, balcony parts in magenta. The planes of overhanging building parts are well distinguishable, but the window planes (if not opened) are very close to its surrounding wall parts. The mirroring and light transmission effects in the window sections lead to geometrically instable plane estimations.

7 CONCLUSION AND OUTLOOK

We presented a novel approach for improving image segmentations for aerial imagery by combining the initial watershed segmentation with information from a 3D point cloud derived from two or three views. For each region, we estimate the most dominant plane, and only the plane parameters are used to trigger the merging process of the regions. With respect to building extraction, our algorithm achieves satisfying results, because the ground and major building structures are better segmented.

In the next steps, we want to search for multiple planes for each region, and we want to implement a splitting routine, so that regions can either get merged or split. If we have such a reliable function, we would start the region merging using the MDL criterion based on the image intensities. So, we can search for geometric descriptions in all, and not only in the big image regions. Furthermore, our approach

can get improved, if we estimate more general geometric primitives for representing the object's surfaces.

Acknowledgements

This work was done within the project *Ontological scales for automated detection, efficient processing and fast visualisation of landscape models*, which is supported by the German Research Foundation (DFG). The authors would also like to thank our student Frank Münster for preparing the data and assisting the evaluation.

REFERENCES

Abraham, S. and Hau, T., 1997. Towards autonomous high-precision calibration of digital cameras. In: SPIE, pp. 82–93.

Binford, T., 1981. Inferring surfaces from images. Artificial Intelligence 17(1-3), pp. 205–244.

Drauschke, M., 2009. An irregular pyramid for multi-scale analysis of objects and their parts. In: GbRPR'09, LNCS 5534, pp. 293–303.

Drauschke, M., Schuster, H.-F. and Förstner, W., 2006. Detectability of buildings in aerial images over scale space. In: PCV'06, IAPRS 36 (3), pp. 7–12.

Fischer, B. and Buhmann, J. M., 2003. Path-based clustering for grouping smooth curves and texture segmentation. PAMI 25(4), pp. 513–518.

Fischler, M. and Bolles, R., 1981. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. ACM 24(6), pp. 381–395.

Förstner, W. and Gülch, E., 1987. A fast operator for detection and precise location of distinct points, corners and centers of circular features. In: ISPRS Conf. on Fast Processing of Photogramm. Data, pp. 281–305.

Gallup, D., Frahm, J.-M., Mordohai, P., Yang, Q. and Pollefeys, M., 2007. Real-time plane-sweeping stereo with multiple sweeping directions. In: CVPR'07.

Guigues, L., Le Men, H. and Cocquerez, J.-P., 2003. The hierarchy of the cocoons of a graph and its application to image segmentation. Pattern Rec. Lett. 24(8), pp. 1059–1066.

Heinrichs, M., Rodehorst, V. and Hellwich, O., 2007. Efficient semi-global matching for trinocular stereo. In: PIA'07, IAPRS 36 (3/W49A), pp. 185–190.

Heuel, S., 2004. Uncertain Projective Geometry. LNCS 3008, Springer.

Hirschmüller, H., 2005. Accurate and efficient stereo processing by semi-global matching and mutual information. In: CVPR, pp. II: 807–814.

Hoffman, D. D. and Richards, W. A., 1984. Parts of recognition. Cognition 18, pp. 65–96.

Kanade, T. and Okutomi, M., 1994. A stereo matching algorithm with an adaptive window: Theory and experiment. PAMI 16(9), pp. 920–932.

Khoshelham, K., 2005. Region refinement and parametric reconstruction of building roofs by integration of image and height data. In: CMRT'05, IAPRS 36 (3/W24), pp. 3–8.

Läbe, T. and Förstner, W., 2006. Automatic relative orientation of images. In: Proc. 5th Turkish-German Joint Geodetic Days.

Lowe, D., 2004. Distinctive image features from scale-invariant keypoints. IJCV 60(2), pp. 91–110.

Martin, D., Fowlkes, C. and Malik, J., 2004. Learning to detect natural image boundaries using local brightness, color, and texture cues. PAMI 26(5), pp. 530–549.

Mayer, H. and Reznik, S., 2005. Building façade interpretation from image sequences. In: CMRT'05, IAPRS 36 (3/W24), pp. 55–60.

Pal, N. R. and Pal, S. K., 1993. A review on image segmentation techniques. Pattern Rec. 26(9), pp. 1277–1294.

Pan, H.-P., 1994. Two-level global optimization for image segmentation. P&RS 49(2), pp. 21–32.

Rissanen, J., 1989. Stochastic Complexity in Statistical Inquiry. World Scientific.

Rottensteiner, F. and Jansa, J., 2002. Automatic extraction of buildings from lidar data and aerial images. In: CIPA, IAPRS 34 (4), pp. 569–574.

Sahoo, P., Soltani, S. and Wong, A., 1988. A survey of thresholding techniques. CVGIP 41(2), pp. 233–260.

Schnabel, R., Wahl, R. and Klein, R., 2007. Efficient ransac for point-cloud shape detection. Computer Graphics Forum 26(2), pp. 214–226.

Sohn, G., 2004. Extraction of buildings from high-resolution satellite data and lidar. In: 20th ISPRS Congress, IAPRS 35 (B3), pp. 1036–1042.

Tao, H. and Sawhney, H. S., 2000. Global matching criterion and color segmentation based stereo. In: Workshop on Applications of Computer Vision, pp. 246–253.

Treisman, A., 1986. Features and objects in visual processing. Scientific American 225, pp. 114–125.

Tuytelaars, T. and Van Gool, L., 2000. Wide baseline stereo matching based on local, affinely invariant regions. In: BMVC, pp. 412–422.

Yang, Q., Wang, L., Yang, R., Stewénius, H. and Nistér, D., 2009. Stereo matching with color-weighted correlation, hierarchical belief propagation, and occlusion handling. PAMI 31(3), pp. 492–504.

Zhao, J. and Lei, S., 2006. Automatic digital image enhancement for dark pictures. In: ICASSP, pp. II: 953–956.

REFINING BUILDING FACADE MODELS WITH IMAGES

Shi Pu and George Vosselman

International Institute for Geo-Information Science and Earth Observation (ITC)
Hengelosestraat 99, P.O. Box 6, 7500 AA Enschede, The Netherlands
spu@itc.nl, vosselman@itc.nl

Commission III/4

KEY WORDS: building reconstruction, data fusion, image interpretation

ABSTRACT:

Laser data and optical data have a complementary nature to 3D features' extraction. Building reconstruction by fusion of the two data sources can reduce the complexity of approaches from either side. In this paper we present a model refinement method, which uses the strong lines extracted from close-range images to improve building models reconstructed from terrestrial laser point clouds. First, model edges are projected from model space to image space. Then significant line features are extracted from an image with Canny edge detector and Hough transformation. Each model edge is then compared with its neighboring image lines to determine the best match. Finally the model edges are updated according to their corresponding image lines. The refinement process not only fixes certain geometry errors of the original models, but also adapts the models to the image data, so that more accurate texturing is achieved.

1 INTRODUCTION

The technique of automated building facade reconstruction is useful to various applications. For urban planning, building facade models provide important references to the city scenes from the street level. For historical building documentation, a large number of valuable structures are contained on the facades, which should be recorded and reconstructed. For all virtual reality applications with users' view on the street, such as virtual tourism and computer games, the accuracy and/or realistic level of the building facade models are vital to successfully simulate an urban environment.

A number of approaches (Dick et al., 2001, Schindler and Bauer, 2003, Frueh et al., 2005, Pollefeys et al., 2008) are available for reconstructing building facades automatically or semi-automatically. Close range image and terrestrial laser point cloud are the commonly used input data. Image based building reconstruction has been researched for years. From multiple 2D images captured from different positions, 3D coordinates of the image features (lines for example) can be calculated. Although acquisition of images is cheap and easy, the difficulties of image understanding make it still difficult to automate the reconstruction using only images. Laser altimetry has been used more and more in recent years for automated building reconstruction. This can be explained by the explicit and accurate 3D information provided by laser point clouds. Researches (Vosselman, 2002, Frueh et al., 2004, Brenner, 2005) suggest that the laser data and images are complementary to each other, and efficient integration of the two data types will lead to a more accurate and reliable extraction of three dimensional features.

In the previous work we presented a knowledge based building facade reconstruction approach, which extracts semantic facade features from terrestrial laser point clouds and combines the feature polygons to water-tight polyhedron models (Pu and Vosselman, 2009). Some modeling errors still exist, and some of them can hardly be corrected by further exploiting the laser data. In this paper, we present a model refinement method which uses strong line features extracted from images to improve the building facade models generated from only terrestrial laser points. The refinement not only fixes the models' geometry errors, but also solves inconsistencies between laser and image data, so that a more accurate texturing can be achieved.

This paper is organized as follows. Section 2 gives an overview of the presented method. Section 3 provides the context research of building reconstruction from terrestrial laser scanning. Section 4 explains the preprocessing steps such as perspective conversion and spatial resection, to make images usable for refining models. Section 5 elaborates the image processing algorithms used for significant line extraction and the matching and refinement strategies. Experiments on three test cases are discussed in section 6. Some conclusions and outlooks are drawn in the final section.

2 METHOD OVERVIEW

A building facade model may contain various errors. For a model reconstructed from terrestrial laser points, the model edges may have certain offset with their actual positions. These errors are caused by gaps in laser points and the limitations of laser data based reconstruction algorithms. Edges are delineated accurately in images. After registering to the model space, image lines can provide excellent reference from which the model edge errors can be fixed. Another necessity of this refinement is to solve the inconsistencies between the laser space and the image space, so that accurate texturing can be achieved.

Before starting the refinement, a 2D image needs to be referenced to the 3D model space, a problem often referred as spatial resection in photogrammetry. We use the standard resection solution of collinearity equations, which requires minimum three image points with their coordinates in model space. To find significant line features from an image, we first detect edges using the Canny algorithm (Canny, 1986), then apply the Hough transform to further extract strong line features from edges. Then model edges are projected to the image space and matched with the image lines. The best match is determined by the geometric properties of candidates and the geometric relations between candidates and the model edge. Finally each model edge with successful matching is projected to the matched image line accordingly, and model edges without any matching are also adjusted to maintain a well shape. Figure 1 gives a flowchart of the refinement process.

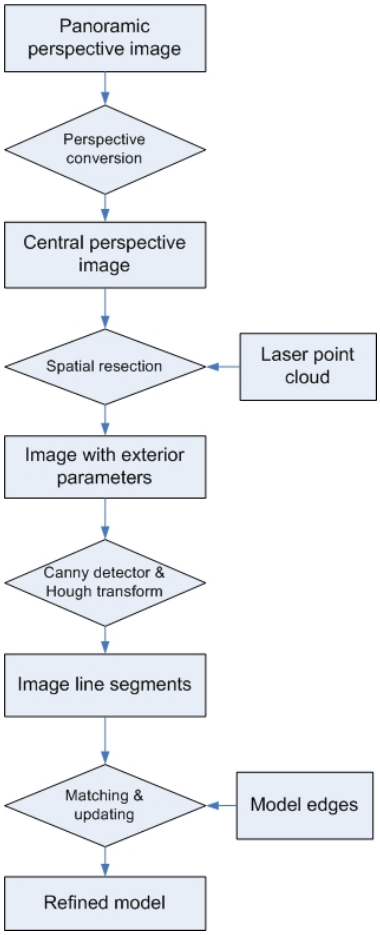


Figure 1: Model refinement process

3 BUILDING RECONSTRUCTION FROM TERRESTRIAL LASER SCANNING

Pu and Vosselman (2009) presents an automatic approach to extract building facade features from a terrestrial point cloud. The method first defines several important building features based on knowledge about building facades. Then the point cloud is segmented to planar segments. Finally, each segment is compared with building feature constraints to determine which feature this segment represents. The feature extraction method works fine for all facade features except for windows, because there are insufficient laser points reflected from window glass. Therefore a hole based window extraction method is introduce. Then polygons to extracted feature segments and the merging of polygons to a complete facade model. An advantage of this approach is that semantic feature types are extracted and linked to the resulting models, so that i) it is possible to get faster visualization by sharing the same texture for same feature type; ii) polygons can be associated with various attributes according to its feature type.

Figure 2 shows a building facade model which is reconstructed with the above approach. The generated building outline seems to coincide with laser points well. However, if we take a close look, it is easy to identify several mistakes from the model. By analyzing more models, we figured two main reasons for the modeling errors. They are:

- Limitations of outline generation method. For example, side wall's eave can "attract" the side boundary edges of the facade, and result in a slight wider polygon in horizontal direction. The almost vertical or horizontal edges are forced to be vertical or horizontal; however, this is not always beneficial.
- Poor scanning quality. Due to the scanning strategy of static laser scanner, complete scanning of a scene seems impossible. There are always some parts which contain very sparse laser points, because of their visibility to any of the scan positions. Occluded zones without any laser points are also usual in laser point clouds. The lack of reference laser information leads to gaps in the final model. For example, the lower part of roofs are hardly scanned because the eaves occlude the laser beams. The directly fitted roof polygons are smaller than their actual sizes. Sometimes these gaps are foreseen and filled using knowledge. For example, we know a roof must attach to the upper side of an eave, so we can extend the roof polygon so that it intersects the eave. However, knowledge based estimation are not always correct.

Figure 2: A reconstructed building facade model, shown together with segmented laser points

4 PREPROCESSING

In order to extract straight lines, an image need to be in central perspective and undistorted. The exterior orientation parameters and focal length should be determined so that 3D model edges can be projected to the same image space for comparison. These are the two objectives of the preprocessing step. An omni-directional panoramic image called Cyclorama is used in our method development, therefore conversion of Cyclorama to central perspective are explained first in 4.1. A semi-automatic approach for exterior orientation calculation is given in 4.2.

4.1 Perspective conversion of Cyclorama

The Cycloramas are created from two fisheye images with a field of view of 185 degree each (van den Heuvel et al., 2007). The camera is turned 180 degree between the two shots. The Cycloramas we used contain image data for the full sphere stored in a panorama image of 4800 by 2400 pixels, corresponding to 360 degree in horizontal direction and 180 degree in vertical direction. Thus, on both directions the angular resolution is 0.075 degree per pixel. With the integrated GPS and IMU devices, all Cycloramas are provided with north direction aligned at x=2400 and horizontal plane aligned at y=1200.

The equiangular projection of the fisheye camera model is described in Schneider and Maas (2003). The projection of Cycloramas to central projective can be understood as projecting a

panoramic sphere part to an assumed plane. First, we make two lines by connect the image acquisition point (perspective center) with the most left and most right model vertices. The angles of the two lines with north direction derive the longitude boundaries of the region of interests (ROI). In practice we widen the ROI to both left and right by 100 pixels, because the image acquisition positions provided by GPS are not so reliable. The principal point is set on the sphere equator, with middle longitude of the two boundaries. Assuming the perspective center coincide in both perspectives, the pixels inside the ROI are converted from panoramic perspective to central perspective according to the following equations:

$$\alpha = \frac{x_p - x_0}{r} \tag{1}$$

$$\beta = \frac{y_p - y_0}{r} \tag{2}$$

$$\tan\alpha = \frac{x_c - x_0}{f} \tag{3}$$

$$\tan\beta = \frac{(y_c - y_0) \times \cos\alpha}{f} \tag{4}$$

where (x_p, y_p) is the pixel coordinate in panoramic perspective; (x_c, y_c) is the pixel coordinate in central perspective; (x_0, y_0) is the principal point; r is the angle resolution; α and β represent the longitude and latitude of the pixel on the panoramic sphere; f is the distance of the panoramic sphere center to the assumed plane, can also be seen as the focal length of the converted central perspective image. With equation 1 to 4 the unique relation between (x_p, y_p) and (x_c, y_c) can be determined.

4.2 Spatial resection

In order to get an unique solution for the six unknown exterior orientation parameters, at least observations of three image control points should be available to form 6 collinearity equations. Figure 3 illustrates the interface for selecting tie points from a laser point cloud and an image. In this implementation it is required to select at least four tie pairs, with one pair for error checking. If more than four pairs are selected, a least squares adjustment is performed to obtain better results.

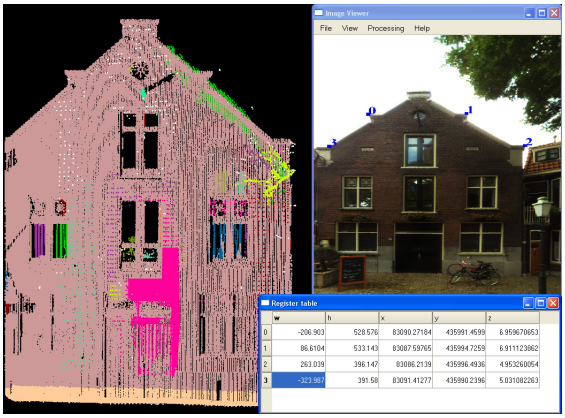


Figure 3: Selecting tie points for spatial resection

5 MODEL REFINEMENT

5.1 Extraction of significant lines from images

The Canny edge detector algorithm (Canny, 1986) is used for initial line extraction (see Figure 4(a) and Figure 4(b)). Here two threshold parameters should be specified for edge linking and finding initial segments of strong edges. Thresholds set too high can miss important information. On the other hand, thresholds set too low will falsely identify irrelevant information as important. It is difficult to give a generic threshold that works well on all images. In addition to the conventional Canny algorithm, we apply a histogram analysis on the image gradients in order to adaptively specify the threshold values. However, factors such as illumination, material, and occlusions still result in many irrelevant edges. In the other hand, some desired edges may not be extracted due to the nature of images. For example, outlines of a wall with very similar color with surrounding environment will not be detected. Outlines inside shadow areas can hardly be extracted either.

Strong line features are further extracted from Canny edges by Hough transformation (see Figure 4(c)). Because of the unpredicted number of edges resulted from the previous step, a lot of irrelevant Hough line segments may also be generated. To minimize the number of these noise lines, instead of adjusting the thresholds of Hough transformation, we sort all the Hough line segments according to their length, and only keep a certain number of longest ones. This is based on the assumption that building outlines are more the less the most significant edges in an image. The limitations of this assumption are already anticipated before applying to practice. For example, large and vivid patterns on a wall's surface can result in more significant line features than the wall edges.

(a) Raw image (b) Canny edges (c) Hough lines

Figure 4: Extracting significant lines from an image

5.2 Matching model edges with image lines

To match model edges and the image lines, both should be located either in the 3D model space or 2D image space. We have chosen the latter space, because projecting object from 3D to 2D is much easier than the other way around. With the calculated exterior orientation parameters from spatial resection and the focal length, model edges can be projected to the image space according to the collinearity equations (see the blue lines in Figure 5).

Assuming a relatively accurate exterior orientation and the focal length are available, the best matched image Hough line for a model edge is determined in two stages:

1. Candidates of best matching image lines are filtered by their parallelism and distance with the model edge (see the green lines in Figure 5). In other words, the angle between a candidate with the model edges should be smaller than a threshold (5 degree for example), and their distance should also be smaller than a threshold (half meter for example). Note the the actual distance threshold is in pixel, which are also "projected" from a 3D distance on the wall plane. If the exterior orientation and focal length are perfect, most model

edges should very well coincide with a strong image line. However, in practice there are often a small offset and an angle between a model edge and its corresponding image line. The optimal angle and distance threshold value are dependent on the quality of exterior orientation parameters and focal length.

2. A best match is chosen from all candidates according to either the collinearity of the candidates or the candidate's length (see the purple lines in Figure 5). It is a common case that a strong line is split to multiple parts by occlusions or shadows. If a number of hough line segments belong to a same line, we set this line as the best match. If not, the longest candidate is chosen as the best match.

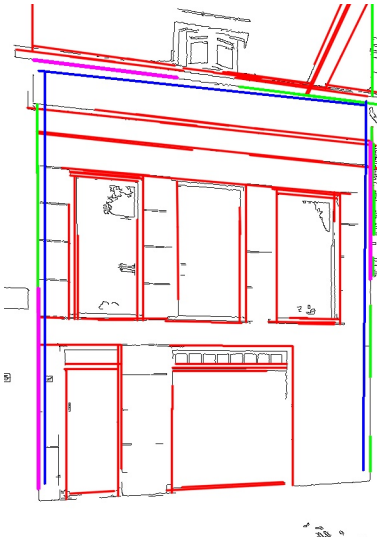


Figure 5: Matching model edges with image lines (Blue: model edges' projection in the image; red: Hough lines; green: candidates; purple: the best matches)

No spatial index is established in the image space to improve the comparison efficiency, because the search space is already localized to a single building facade, which includes only dozens of edges and Hough lines.

A limitation of this matching method is that it can hardly determine the correct corresponding edge if too many similar line features are within searching range. Simply comparing the geometry properties of position, direction and length are not sufficient in this case. For example, the eaves in Figure 5 result in many significant lines and they are all parallel and close to the wall's upper boundary edges. These eave lines can be distinguished if the eave is also reconstructed and included in the facade model, but ambiguity caused by pure color pattern is still difficult to be solved.

5.3 The refinement strategy

After matching, most model edges should be associated with a best matched image line. These model edges are updated by projecting to their best matched image line. There are some model edges which don't match any image lines. If no change is made to an edge with its previous or next edge changed, strange shapes like sharp corners and self-intersections may be generate. Therefore interpolations of the angle and distance change from the previous and next edges, are applied to the edges without matched image lines. With these refinement strategies, an original model is updated to be consistent with the geometry extracted from images, and the model's geometry validity and general shape are also maintained.

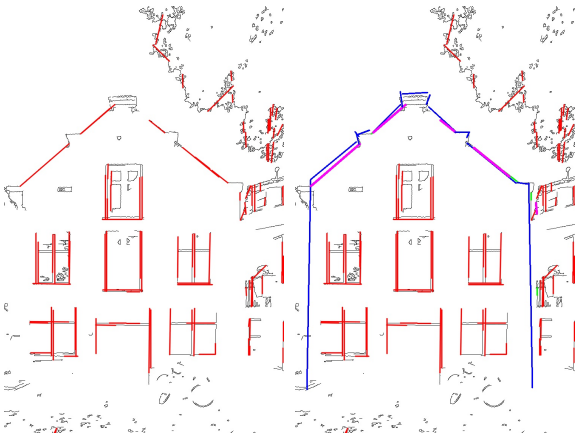
Finally, the refined model edges in image space need to be transferred back to the model space. Because the model edges are only moved on their original 3D planes, which is known, the collinearity equations are used again to calculate the new 3D positions of all the modified model vertices.

6 TEST CASES

In this section, three data sets are experimented with the presented refinement method. The building models are produced with the reconstruction approach introduced in Section 3. All the images are originally provided as Cycloramas. The central perspective conversion and exterior orientation calculation follow the processes explained in Section 4.

6.1 The restaurant house

The inconsistencies between the model edges and image lines in Figure 6 are mainly due to inaccurate exterior orientation of the image. It is difficult to pick an image point accurately by manual operation. Picking the corresponding point in a laser point cloud is also a difficult job. Automated texturing of building facade models is desired in the context of our research. The quality of the exterior orientation is a key issue to the texturing effect. Even a minor inaccuracy in the exterior orientation parameters can lead to poor texture result, as shown in Figure 7(a). Applying our refinement method, several model edges are linked with their matched image lines (see Figure 6(b)), and are updated accordingly. The texture result is significantly improved as shown in Figure 7(b), with the sky's background color removed. However, the middle top part of the facade model is still not refined, because this image part is too blurred to output a Hough line.



(a) Hough lines (b) Matching

Figure 6: Matching model edges with image lines for refining the restaurant house's model

6.2 The town hall

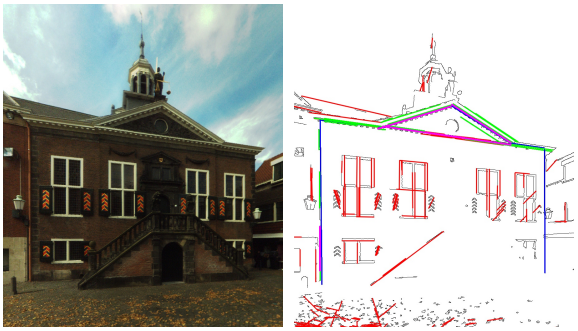
The upper boundary of the town hall in Figure 8(a) contains a lot of tiny details, which are well recorded by laser scanning and modeled as sawtooth edges in the building facade model. Instead of adjusting the outline generation parameters in the reconstruction stage, we can also use the presented image based refinement to smooth the model outline. Figure 8(b) shows the matching step. The model's upper edges are successfully matched to the strong lines, which actually come from the eave. In this example

(a) Before refinement (b) After refinement

Figure 7: Comparison of textured restaurant house model before and after refinement

the wall's upper boundary is not detected in the image, because it is occluded from sunlight by the eaves when the image was taken.

The left boundary of the building model is modeled correctly from laser points by intersecting two large wall planes. However, in practice it is matched to a strong contrast caused by a water pipe on the wall (see Figure 8(b)). This again, reveals the limitations of this refinement method.



(a) Raw image (b) Matching

Figure 8: Matching model edges with image lines for refining the town hall's model

6.3 The wall with high windows

In this example, the refinement is applied to improve the windows extracted from the holes from laser points of a wall (Pu and Vosselman, 2009). The contrast of a window and its surrounding wall are usually rather obvious in optical data. Strong line features are frequently found at the windows' boundaries and frames, and can be used to refine the information from the laser altimetry. Figure 9(b) shows the window rectangles extracted from laser points, which contain a lot of errors due to limitation of segmentation and modeling algorithms. In Figure 9(a) we apply the same matching stretchy for refinement purpose, and the final result in shown in Figure 9(c). Most windows' boundaries are well corrected according to the image lines. The second left window in the upper row is not improved, because the difference between the modeled shape and the actual shape is too large to correlate them. There are some remaining errors, such as the first, third and sixth window (from left to right) in the lower row. This is because the parameters of Hough transform are too strict to generate any candidate line.

The textured final model after manual adjustment is shown in Figure 10. Without the refinement, 42 vertices need to be manually adjusted. Only 10 vertices need to be adjusted after the refinement. Note that the window rectangles are intruded inside the wall plane, and the intrusion offset is derived from laser points reflected from window glass and window frames.

(a) Matching

(b) Before refinement

(c) After refinement

Figure 9: Matching and refining window boundaries

6.4 Summary

The effectiveness as well as limitations of our refinement method are examined through the three test cases. We realize that the refining effect relies on the following prerequisites:

- Accurate exterior and interior orientations. In particular, the selected tie points for spatial resection should be sufficient (four or more), and should be distributed equally in both horizontal and vertical directions to minimize the computation error.
- No large occlusions in front of the building facade.

Figure 10: A textured building facade model with intruded windows

- Stronger contrast by geometries than optical factors (illumination, color pattern, etc.).

Some limitations of the current refinement method have been located at:

- It cannot solve ambiguities caused by multiple lines with similar geometry properties.
- It cannot distinguish whether a model-image inconsistency is caused by reconstruction errors or inaccurate exterior orientation.

Knowledge based reasoning of the image information is the key to the first problem. The current matching stretchy is rather local. Experiments show that the offset direction between the model edges and their matched image lines are mostly same, which is obviously caused by inaccurate exterior orientation. A globe matching process (RANSAC over offsets for example) should be able to estimate the correct exterior orientations.

7 CONCLUSIONS AND OUTLOOK

In this paper we present a model refinement method, which uses the lines extracted from close-range images to improve building models reconstructed from terrestrial laser point clouds. With the refinement, several modeling errors caused by either gaps in laser data or reconstruction algorithm, are corrected with image information. Texturing is also improved after the refinement.

Nowadays it is more and more common for acquisition platforms to acquire laser data and optical data simultaneously. Line extraction from images is very accurate, while laser points are more suitable to extract planar features. Efficient fusing of laser points and image naturally avoids many barriers for building reconstruction from either sides. The attempt through our refinement method shows promising future for automated building reconstruction by fusing laser altimetry and optical methods.

Two directions of the future work: knowledge based image reasoning and global matching, have been suggested earlier. Besides, nowadays the mainstream image acquisition systems usually determine exterior orientations via GPS and IMU, but they are not accurate. If we use the laser points as reference data, and match image lines with model edges from laser point clouds (similar to this research), there should be enough control points for estimating the accurate exterior orientations for images.

ACKNOWLEDGEMENTS

The authors would like to thank Cyclomedia B.V. for providing the Cyclorama data.

References

Brenner, C., 2005. Building reconstruction from images and laser scanning. International Journal of Applied Earth Observations and Geoinformation 6(3-4), pp. 187–198.

Canny, J., 1986. A computational approach to edge detection. IEEE Transactions on pattern analysis and machine intelligence pp. 679–698.

Dick, A., Torr, P., Ruffle, S. and Cipolla, R., 2001. Combining single view recognition and multiple view stereo forarchitectural scenes. In: Eighth IEEE International Conference on Computer Vision, 2001. ICCV 2001. Proceedings, Vol. 1.

Frueh, C., Jain, S. and Zakhor, A., 2005. Data processing algorithms for generating textured 3D building facade meshes from laser scans and camera images. International Journal of Computer Vision 61(2), pp. 159–184.

Frueh, C., Sammon, R. and Zakhor, A., 2004. Automated texture mapping of 3D city models with oblique aerial imagery. In: 3D Data Processing, Visualization and Transmission, 2004. 3DPVT 2004. Proceedings. 2nd International Symposium on, pp. 396–403.

Pollefeys, M., Nister, D., Frahm, J., Akbarzadeh, A., Mordohai, P., Clipp, B., Engels, C., Gallup, D., Kim, S., Merrell, P. et al., 2008. Detailed real-time urban 3d reconstruction from video. International Journal of Computer Vision 78(2), pp. 143–167.

Pu, S. and Vosselman, G., 2009. Knowledge based reconstruction of building models from terrestrial laser scanning data (in press). ISPRS J. Photogramm. Remote Sens.

Schindler, K. and Bauer, J., 2003. A model-based method for building reconstruction. In: First IEEE International Workshop on Higher-Level Knowledge in 3D Modeling and Motion Analysis, 2003. HLK 2003, pp. 74–82.

Schneider, D. and Maas, H., 2003. Geometric modelling and calibration of a high resolution panoramic camera. Optical 3D Measurement Techniques VI (Eds. Gruen, A. and Kahmen, H.) 2, pp. 122–129.

van den Heuvel, F., Verwaal, R. and Beers, B., 2007. Automated Calibration of Fisheye Camera Systems and the Reduction of Chromatic Aberration. Photogramm. Fernerkund. Geoinf. 3, pp. 157.

Vosselman, G., 2002. Fusion of laser scanning data, maps, and aerial photographs for building reconstruction. In: 2002 IEEE International Geoscience and Remote Sensing Symposium, 2002. IGARSS'02, Vol. 1.

AN UNSUPERVISED HIERARCHICAL SEGMENTATION OF A FAÇADE BUILDING IMAGE IN ELEMENTARY $2D$ - MODELS

Jean-Pascal Burochin, Olivier Tournaire and Nicolas Paparoditis

Université Paris-Est, Institut Géographique National, Laboratoire MATIS
73 Avenue de Paris, 94165 Saint-Mandé Cedex, France
{firstname.lastname}@ign.fr

Commission III/3, III/4

KEY WORDS: street level imagery, façade reconstruction, unsupervised hierarchical segmentation, gradient accumulation, recursive split, model matching

ABSTRACT:

We introduce a new unsupervised segmentation method adapted to describe façade shapes from a single calibrated street level image. The image is first rectified thanks to its vanishing points to facilitate the extraction of façade main structures which are characterized by a horizontal and vertical gradient accumulation which enhances the detection of repetitive structures. Our aim is to build a hierarchy of rectangular regions bounded by the local maxima of the gradient accumulation. The algorithm recursively splits horizontally or vertically the image into two parts by maximizing the total length of *regular edges* until the radiometric content of the region hypothesis corresponds to a given model (planar and generalized cylinders). A *regular edge* is a segment of a main gradient direction that effectively matches to a contour of the image. This segmentation could be an interesting tool for façade modelling and is in particular well suited for façade texture compression.

1 INTRODUCTION

1.1 Context

Façade analysis (detection, understanding and reconstruction) from street level imagery is currently a very active research domain in the photogrammetric computer vision field. Indeed, it has many applications. Façade models can for instance be used to increase the level of details of $3D$ city models generated from aerial or satellite imagery. They are also useful for a compact coding of façade image textures for streaming or for an embedded system. The characterization of stable regions in façades is also necessary for a robust indexation and image retrieval.

1.2 Related work

Existing façade extraction frameworks are frequently specialized for a certain type of architectural style or a given texture appearance. In a procedural way, operators often step in a pre-process to split correctly the image in suitable regions. Studied images indeed are assumed to be framed in such a way that they exactly contain relevant information data such as windows on a clean wall background.

Most building façade analysis techniques try to extract specific shapes/objects from the façade: windows frame, etc. Most of them are data driven, *i.e.* image features are first extracted and then some models are matched with them to build object hypotheses. Some other model-driven techniques try to find more complex objects which are patterns or layouts of simple objects (*e.g.* alignments in $1D$ or in $2D$). Higher level techniques try to generate directly a hierarchy of complex objects composed of patterns of simple objects usually with grammar-based approaches. Those methods generally devote their strategy to a special architectural style.

1.2.1 Single pattern detection Strategies to extract shape hypotheses abound in recent works. (Čech and Šára, 2007), for instance, propose a segmentation based on a maximum *a posteriori* labeling. They associate each image pixel with values linked with some configuration rules. They extract a set of non-overlapping windowpanes hypotheses, assumed to form axis-parallel rectangles of relatively low variability in appearance. This restriction does not take into account lighting variations. With a supervised classification-based approach, (Ali et al., 2007) extracted windows width an *adaboost* algorithm. In the same fashion, (Wenzel and Förstner, 2008) minimize user interaction with a clustering procedure based on appearance similarity.

Assuming the regularity of the façade, (Lee and Nevatia, 2004) use a gradient profile projection to locate window edges coordinates. They first locate valley between two extrema blocks of each gradient accumulation profile and they roughly frame some floors and windows columns. Edges are then adjusted on local data information. Their results are relevant for façades whose background does not contain any contours such as railings, balconies or cornices.

1.2.2 $1D$ or $2D$ grid structures detection (Korah and Rasmussen, 2007, Reznik and Mayer, 2007) use linear primitives to generate rectangle hypotheses for windows. A *Markov Random Field* (*MRF*) is then used to constrain the hypotheses on a $2D$ regular grid. (Korah and Rasmussen, 2007) generate their rectangular hypotheses in a similar way as (Han and Zhu, 2005): they project on image $3D$ planar rectangles. (Reznik and Mayer, 2007) learn windows outline from training data and use as hypotheses for window corners characteristic points.

1.2.3 Façade grammars A façade grammar describes the spatial composition rules of complex objects (*e.g.* grid structure) and/or simple objects to construct a façade. Approaches based on grammars succeed in describing only façades corresponding to the grammar. Nevertheless, to obtain a detailed description a specific grammar is required per type of architecture (*e.g.* Hausmanian in the case of Parisian architecture). The drawback is that many grammars are necessary to describe the variety of building architectures in a general framework.

For instance, to detect windows on simple buildings, (Han and Zhu, 2005) integrates rules to produce patterns in image space. In particular, this approach integrates a bottom-up detection of rectangles coupled with a top-down prediction hypotheses taken from the grammar rules. A Bayesian framework validates the process. (Alegre and Dellaert, 2004) look for rectangular regions with homogeneous aspect by computing radiometry variance. (Müller et al., 2007) extract an *irreducible region* to summarize the façade by periodicity in vertical and horizontal directions. Their results are significant with façades that effectively contain regular window grid pattern or suitable perspective effects. (Ripperda, 2008) fixes her grammar rules according to prior knowledge: she beforehand computes distribution of façade elements from a set of façade images.

These approaches either use a too restrictive model dedicated to simple façade layout, or are too specialized for a particular kind of architecture. They thus would hardly deal with usual Parisian façades such as Hausmanian buildings or other complex architectures with balconies or decoration elements.

Our process works exclusively on a single calibrated street-level image. Although we could have, we voluntarily did not introduce additional information such as $3D$ imagery (point clouds, etc.) because for some applications such as indexation, image retrieval and localization, we could just have a single photo acquired by a mobile phone.

2 OUR MODEL BASED SEGMENTATION STRATEGY

Most of the aforementioned approaches provide good results for relatively simple single building. Only a few of them have addressed very complex façade networks such as the ones encountered in European cities where the architectural diversity and complexity is large. Our work is upstream from most of these approaches: we do not try to extract semantic information but we just propose a façade segmentation framework that could be helpful for most of these approaches. This framework must firstly separate a façade from its background and neighboring façades, and then, identify intra-façade regions of specific elementary texture models. These regions must be robust to change in scale or point of view.

Our strategy requires horizontal and vertical image contour alignments. We thus first need to rectify images in the façade plane: vertical and horizontal directions in the real world respectively become vertical and horizontal in the image. To do so, we extract vanishing points which provide an orthogonal basis in object space useful to resample the image as required.

Regarding segmentation, the core of our approach relies on a recursive split process and a model based analysis of each subdivided regions. Indeed we do not intend to directly match a model to the whole façade, but we build a tree of rectangular regions by recursively confronting data with some basic models. If a region does not match with any of them, it is split again, and the two sub-regions are analyzed as illustrated by the decision tree on figure 1. Our models are based on simple radiometric criteria: planes and generalized cylinders. Such objects are representative of frequent façade elements like window panes, wall background or cornices.

We start each process with the whole image region. We test if its texture matches our planar model. If it does, then the process stops: we have recognized a planar and radiometrically coherent region in the image. Otherwise, we test if it matches our generalized cylinder model. In the same manner, the process stops

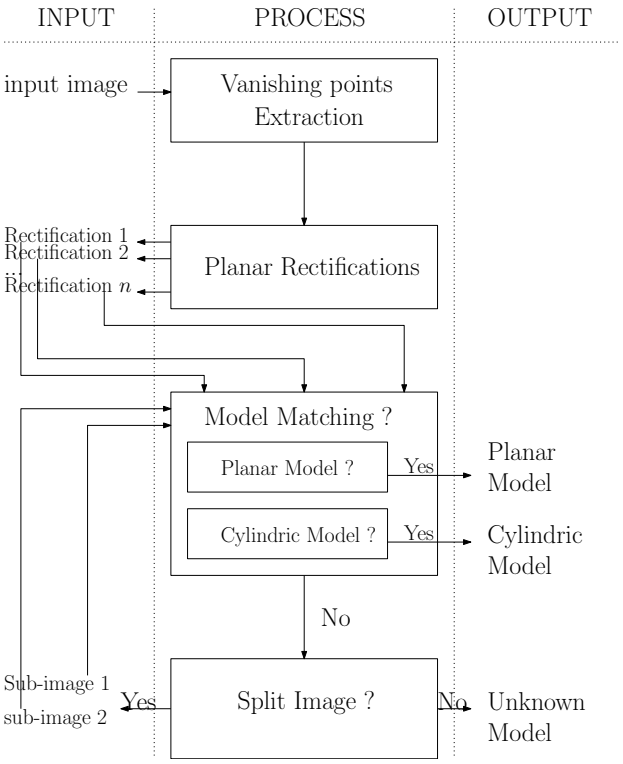


Figure 1: Our algorithm recursively confronts data with models. If region does not match with any proposed model, we split it.

on the cylinder model. Otherwise the region is not considered as homogeneous (in the sense of our models) and it is split in two sub-regions. The process recursively analyzes these two sub-regions exactly as the same way as the large region. Thus, we build a segmentation tree whose leaves are planar or generalized cylinder models. The following sections explain each step of this algorithm.

3 RECTIFICATION PROCESS

3.1 Extracting Vanishing Points

Our rectification process relies on vanishing point lines detected by (Kalantari et al., 2008). They project relevant image segments on the Gaussian sphere: each image segment is associated with a point on the sphere. Their algorithm relies on the fact that each circle of such a $3D$-point distribution gathers points associated with the same vanishing point in the image. Then they estimate the best set of circles that contains the highest number of points. The more representative circles are assumed to provide main façade directions: the vertical direction and several horizontal ones. Figure 2 upper-right shows some detected edges that support main vanishing points: segments associated with the same direction are drawn in the same color.

3.2 Multi-planar Rectification Process

We rectify our image in each plane defined by a couple of one of the horizontal vanishing points and the vertical one. We then project the image onto the plane. Figure 2 bottom right shows a rectification result. Figure 2 bottom left shows rectified edges on the façade plane.

Calibration intrinsic parameters are supposed to be known. Rectified image is resampled in grey levels, but such a restriction already provides some interesting perspectives.

Figure 2: The rectification process. upper left: original image only; upper right: original image with segments that support main vanishing points (green and blue ones are those for the main vertical direction, yellow and white ones for the main horizontal direction and red ones for an aberrant vanishing point); bottom left: rectified image with rectified segments that support the two selected directions; bottom right: rectified image only

4 MODEL MATCHING

Given a region of a rectified image, we try to match two geometric models with data in increasing complexity order: the planar model $\mathcal{M}_1$, then the generalized cylinders one $\mathcal{M}_2$. This decision tree indeed provides a good compromise between quality and compression rate.

To match an image region with a model, we simply count local radiometric differences as follows. Let I_k be the sub-image at region R_k of a façade image I. Sub-image I_k is described by model $\mathcal{M}$ when the deviation $N_{\mathcal{M}}(I_k)$ is small enough and if this model is the simplest one. Deviation $N_{\mathcal{M}}(I_k)$ is defined by the number of pixels whose radiometry differs too much from the model. Radiometric medians provide some significative robustness: the influence of parasite structures such as tree branches or lighting posts, is significantly reduced. Figure 3 illustrates models we use.

4.1 Planar Model

A planar model is an image with an uniform radiometry. Let $\mathcal{M}_1$ be the planar model of a sub-image I_k. It is defined by equation 1.

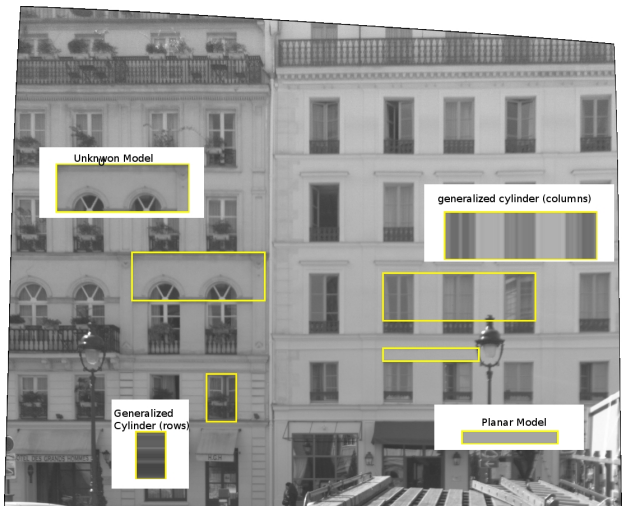


Figure 3: Description of our radiometric $2D$-models

An instance of a planar model is depicted on the lower-right corner of figure 3.

$$\mathcal{M}_1 : \forall p \in R_k, I_k(p) = median(I_k) + \epsilon(p) \qquad (1)$$

where $\epsilon(p)$ is the difference between the image I_k and the model $\mathcal{M}_1$ at the pixel p. If this difference is smaller than an arbitrary threshold, it is tolerated. It refers to the acquisition noise or some texture defects. Otherwise, the deviation $N_{\mathcal{M}}(I_k)$ is incremented.

4.2 Generalized Cylinder Model

A generalized cylinder model is designed either in columns ($\mathcal{M}_2^c$) or in rows ($\mathcal{M}_2^l$). The model in columns is composed of medians of columns and the cylinder model in rows is composed of medians of rows. They are is defined by equation 2. Functions $median_x$ and $median_y$ respectively return the median of the column at x abscissa and the row at y ordinate. Figure 3 shows an instance of each generalized cylinder model.

$$\begin{array}{l} \forall (x,y) \in R_k, \\ \mathcal{M}_2^c : I_k(x,y) = \quad median_x(I_k(x,y)) + \epsilon(x,y) \\ \mathcal{M}_2^l : I_k(x,y) = \quad median_y(I_k(x,y)) + \epsilon(x,y) \end{array} \qquad (2)$$

where $\epsilon(x,y)$ is the difference between the image I_k and the model $\mathcal{M}_2$ at the pixel (x,y). In the same manner as planar model, the deviation $N_{\mathcal{M}}(I_k)$ is incremented when this difference is greater than an arbitrary threshold.

5 SPLIT PROCESS BY ENERGY MAXIMIZATION

Given a region of a rectified image that does not match with any model, we try to split it by measuring the internal gradient distribution energy.

5.1 Generating splitting hypotheses

We select split hypotheses with a technique close to (Lee and Nevatia, 2004). We accumulate x-gradient absolute values by column and y-gradient absolute values by row, where x-gradient and y-gradients are related respectively to vertical and horizontal

edges. We use convolution with a discrete first order derivative operator. Local extrema of these accumulations are our split hypotheses. This reinforces low but repetitive contrasts. Figure 4 illustrates this process.

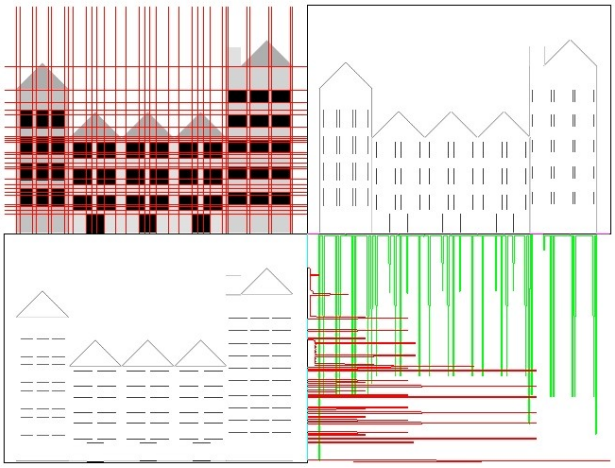


Figure 4: Upper right and bottom left images respectively are x-gradient and y-gradient. Bottom right image presents accumulation profiles: green profile for x-gradient and red one for y-gradient. extrema of this profiles are our split hypotheses: red lines in upper left image.

Such a rough set of hypotheses supplies initial interesting segmentation. (Lee and Nevatia, 2004) base their window detection on similar rough segmentation. They almost use the same procedure except that they do not accumulate gradients in the same orientation: they respectively treat x-gradient and y-gradient horizontally and vertically. Thus they locate valley between two extrema blocks of each gradient accumulation profile and they frame some floors and windows columns. Their results were relevant in façades composed of a fair windows grid-pattern distribution on a clean background.

Main buildings structure are detected. Each repetitive objects are present in vertical or horizontal alignment as common edges generate local extrema in accumulation profiles. Local gradient extremum neighborhood is set *a priori*. In our case, this neighborhood is set to 30 centimeters. However this last grid-pattern usually is not enough by itself to summarize façade texture: repetitive elements of our images are not necessarily evenly distributed. Thus our split strategy relies on breaks between two façades or inside one façade.

5.2 Choosing the best splitting hypotheses

The best splitting hypothesis maximizes its pixels number of *regular edges* in each of the two sub-region. A *regular edge* is a segment of a main gradient direction that effectively matches to a contour of the image. The weight W_H of the split hypothesis H that provides the two regions R_1 and R_2 is given by $W_H = f(R_1) + f(R_2)$, where the function f returns the pixels number of *regular edges* in a region. We select the hypothesis $H^* = \arg\max_H W_H$.

If we try for instance to split image at x_0 location, we reaccumulate y-gradients in left region and in right region separately. Local extrema are detected in each of those y-profiles (cf figure 5).

Previous vertical split hypotheses and those new horizontal split hypotheses constitute two new grid patterns for local split hypotheses. Each edge of these grid patterns is either *regular* or

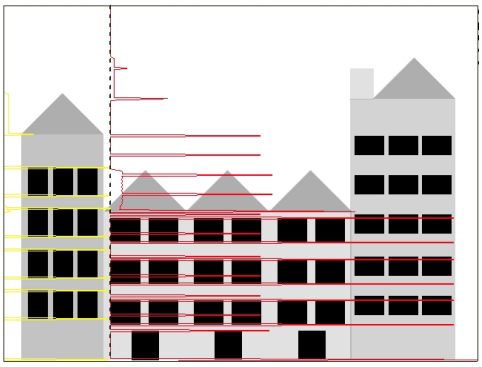


Figure 5: y-gradient profiles are separately accumulated in left region (yellow profile) and in right region (red profile).

fictive. *Regular edges* are located on *significant gradient* where a *significant gradient* keeps its orientation uniform. A *fictive edge* does not match with any significant gradient. Such a distinction is illustrated in figure 6.

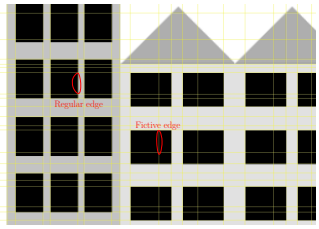


Figure 6: *Regular edges* are located on *significant gradient* where a *significant gradient* keeps its orientation uniform. A *fictive edge* doesn't match with any significant gradient.

The weight of each split hypothesis is the sum of regular edge lengths. Figure 7 illustrates best split selection.

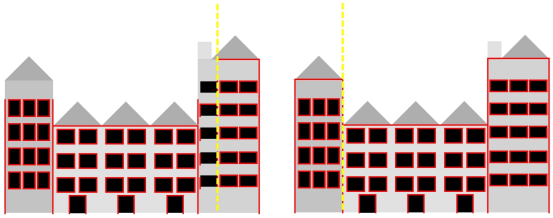


Figure 7: Regular edges are drawn in red. Split hypotheses are drawn in yellow. Right image presents the best split hypothesis whose weight is 8400 regular edge pixels. Left image presents a bad split hypothesis: only 7700 regular edge pixels.

If the given region does not contain any gradient extremum, the process stops. Figure 3 shows a region that do not fit with any model and that is not split.

6 RESULTS

We illustrate our segmentation on a typical instance of our issue: two building façades in the background. We have set maximum model deviation at 15% of each region area. On our images, the depth in the hierarchy of the segmentation tree is represented by the thickness of split lines. First the process detects vertical structure discontinuities (figure 8). The two façades are separated. Then on each of these two new sub-images, background is separated from the foreground (figure 9). At this step we have obtained four images: two façade images and two images of foreground cars. Then the process recursively keeps analyzing these images as shown in figure 10. Figure 11 shows the global segmentation: a tree of about 2000 elementary models.

Figure 8: The two façades are separated because of the significative break between their radiometric structure.

Figure 10: The process recursively segments each of the four sub-images. It splits the two façades and the foreground cars.

Figure 9: Background is separated from the foreground on each of the two façade images.

Figure 11: The segmentation result is a tree of 2000 elementary models.

The strength of this process is its ability to localize accurate global structure breaks: it separates façades and foreground. On the one hand, split results at the foreground are not really interesting because the related region is not in the rectified plane: they are based on chaotic gradient distribution. In such a case, the process stops or it oversegments. This phenomenon typically occurs on the cars of figure 11. On the other hand, splits inside façade texture provides some significative information. On figure 10, the left façade is first split between the second and the third floor, whereas the first windows column is extracted from the right façade. This different strategy certainly must be explained by the fact that the process is exclusively based on edges alignment. An other criterion like contour uniformity may direct the split decision toward a more significant separation: favoring floor separation rather than window columns.

Figure 3 shows the region models, the leaves of the segmentation tree. One can see that the synthetic image reconstructed from the $2D$-models is very close to the initial image although the representation is very compact. This shows that our modelling is particularly well adapted for image compression.

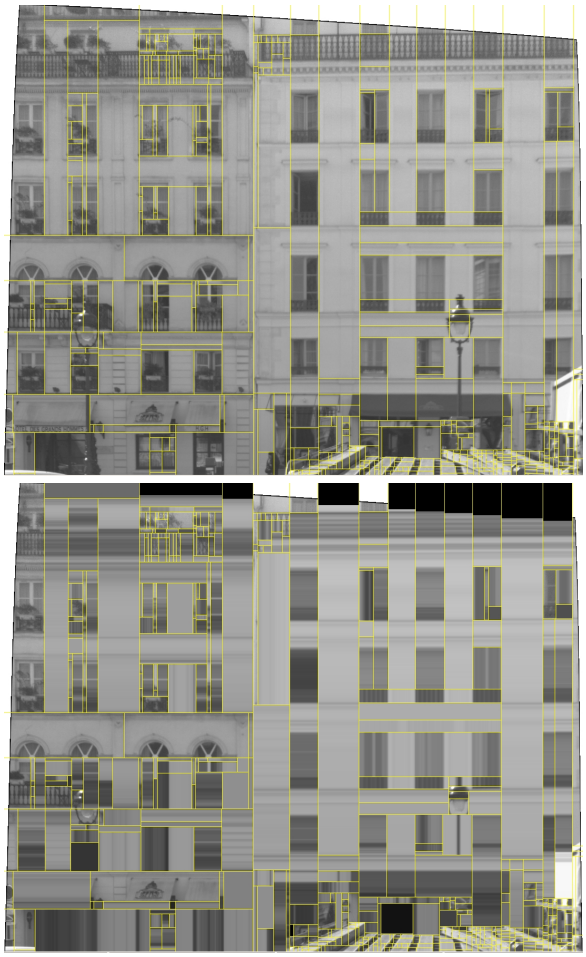


Figure 12: Upper: Rectified façade image. Bottom: Synthetic image reconstructed from 1000 elementary $2D$-models.

7 CONCLUSIONS AND FUTURE WORK

In this paper, we have presented a new unsupervised model-based segmentation approach that provides interesting result. It is able to separate a façade from its surroundings but also to organize façade itself in a hierarchy. Still these are first results, thus there are many improvements that could be made. The dictionary of models is currently being extended to periodic textures to manage for instance balconies, building floors or brick texture. Some other objects or specializations of objects could be added such as symmetry computation of (Van Gool et al., 2007). A merger process at each step of the process could also be useful to correct oversegmentations. Besides we could add color information to directly detect difference between two façades ore between two floors in certain cases. We could also use a point cloud to compute an ortho image: displacements due to perspective effects would be avoided.

Such an unsupervised segmentation will provide of course relevant clues to classify the façade architectural style or to detect objects backward or in front of it. It is also intended to give geometrical information that represents relevant indexation features *e.g.* windows gab length or floor delineation.

REFERENCES

Alegre, F. and Dellaert, F., 2004. A Probabilistic Approach to the Semantic Interpretation of Building Facades. Technical report, Georgia Institute of Technology.

Ali, H., Seifert, C., Jindal, N., Paletta, L. and Paar, G., 2007. Window Detection in Facades. In: Proc. of the 14th International Conference on Image Analysis and Processing, pp. 837–842.

Han, F. and Zhu, S.-C., 2005. Bottom-up/top-down image parsing by attribute graph grammar. In: International Conference on Computer Vision, IEEE Computer Society, pp. 1778–1785.

Kalantari, M., Jung, F., Paparoditis, N. and Guédon, J.-P., 2008. Robust and Automatic Vanishing Points Detection with their Uncertainties from a Single Uncalibrated Image, by Planes Extraction on the Unit Sphere. In: IAPRS, Vol. 37 (Part 3A), Beijing, China.

Korah, T. and Rasmussen, C., 2007. 2D Lattice Extraction from Structured Environments. In: International Conference on Image Analysis and Recognition, Vol. 2, pp. 61–64.

Lee, S. C. and Nevatia, R., 2004. Extraction and Integration of Window in a 3D Building Model from Ground View images. In: Computer Society Conference on Computer Vision and Pattern Recognition, Vol. 2, pp. 113–120.

Müller, P., Zeng, G., Wonka, P. and Van Gool, L., 2007. Image-based Procedural Modeling of Facades. In: Proceedings of ACM SIGGRAPH 2007 / ACM Transactions on Graphics, Vol. 26number 3, p. 85.

Reznik, S. and Mayer, H., 2007. Implicit Shape Models, Model Selection, and Plane Sweeping for 3D Facade Interpretation. In: Photogrammetric Image Analysis, p. 173.

Ripperda, N., 2008. Determination of Facade Attributes for Facade Reconstruction. In: Proc. of the 21st Congress of the International Society for Photogrammetry and Remote Sensing.

Čech, J. and Šára, R., 2007. Windowpane Detection based on Maximum Aposteriori Probability Labeling. Technical report, K13133 FEE Czech Technical University, Prague, Czech Republic.

Van Gool, L. J., Zeng, G., Van den Borre, F. and Müller, P., 2007. Towards Mass-Produced Building Models. In: Photogrammetric Image Analysis, p. 209.

Wenzel, S. and Förstner, W., 2008. Semi-supervised Incremental Learning of Hierarchical Appearance Models. In: Proc. of the 21st Congress of the International Society for Photogrammetry and Remote Sensing.

GRAMMAR SUPPORTED FACADE RECONSTRUCTION FROM MOBILE LIDAR MAPPING

Susanne Becker, Norbert Haala

Institute for Photogrammetry, University of Stuttgart
Geschwister-Scholl-Straße 24D, D-70174 Stuttgart
forename.lastname@ifp.uni-stuttgart.de

KEY WORDS: Architecture, Point Cloud, Urban, LiDAR, Facade Interpretation

ABSTRACT:

The paper describes an approach for the quality dependent reconstruction of building facades using 3D point clouds from mobile terrestrial laser scanning and coarse building models. Due to changing viewing conditions such measurements frequently suffer from different point densities at the respective building facades. In order to support the automatic generation of facade structure in regions where no or only limited LiDAR measurements are available, a quality dependent processing is implemented. For this purpose, facades are reconstructed at areas of sufficient LiDAR point densities in a first processing step. Based on this reconstruction, rules are derived automatically, which together with the respective facade elements constitute a so-called facade grammar. This grammar holds all the information that is necessary to reconstruct facades in the style of the given building. Thus, it can be used as knowledge base in order to improve and complete facade reconstructions at areas of limited sensor data. Even for parts where no LiDAR measurements are available at all synthetic facade structures can be hypothesized providing detailed building geometry.

1. INTRODUCTION

Due to the growing need for visualization and modelling of 3D urban landscapes numerous tools for the area covering production of virtual city models were made available, which are usually based on 3D measurements from airborne stereo imagery or LiDAR. This airborne data collection, which mainly provides the outline and roof shape of buildings, is frequently complemented by terrestrial laser scanning (TLS). However, the applicability of standard TLS is usually limited to the 3D data capturing of smaller scenes from a limited number of static viewpoints. In contrast, the application of dynamic TLS from moving platforms allows the complete coverage of spatially complex urban environments from multiple viewpoints. One example of such a mobile mapping system, which combines terrestrial laser scanners with suitable sensors for direct georeferencing, is the StreetMapper system (Kremer and Hunter, 2007). This system enables the rapid and area covering measurement of dense 3D point clouds by integrating four 2D laser scanners with a high performance GNSS/inertial navigation system. By these means accuracy levels better than 30mm have been demonstrated for point measurement in urban areas (Haala et al., 2008).

In general, such systems allow for an efficient measurement of larger street sections including the facades of the neighbouring buildings. However, depending on the look angle during the scanning process, strong variations of the available point densities at the building facades can occur. Such viewpoint limitations and occlusions will subject the collected point cloud to significant changes of accuracy, coverage and amount of detail. For this reason, the following interpretation of the measured point clouds will be hampered by considerable changes in data quality. Thus, algorithms for automatic facade reconstruction have to be robust against potentially incomplete data sets of heterogeneous quality. For this purpose, dense point cloud measurements for facades with good visibility are used in our approach to extract rules on dominant or repetitive features as well as regularities. These rules then are used as knowledge base to generate facade structure for parts or buildings where no sensor data is available. By these means bottom-up and top-down propagation of knowledge can be combined in order to profit from both reconstruction techniques. The production rules, which are automatically inferred from well observed and modelled facades, are represented by a formal grammar.

Such formal grammars are frequently used within knowledge based object reconstruction to ensure the plausibility and the topological correctness of the reconstructed object elements. Lindenmayer-systems (L-systems), which can be applied to model the growth processes of plants, are well known examples of formal grammars (Prusinkiewicz and Lindenmayer, 1990). So-called split grammars are introduced by Wonka et al. (2003) to automatically generate architectural structures from a database of rules and attributes. Similarly, Müller et al. (2006) present a procedural modelling approach for the generation of detailed building architecture in a predefined style. However, the variety of facade structures which can be generated is restricted to the knowledge base inherent in the grammar rules or model libraries. Thus, the appearance of facade elements is limited to prespecified types, even when leaving some freedom in the values of their parameters. Another problem while applying such approaches for object reconstruction is that manual interaction is required to constitute suitable building styles and translate them into some kind of model or grammar description. For this reason, several approaches aim at deriving such kind of knowledge from observed or given data. For example, Ripperda (2008) derives prior facade information from a set of facade images in order to support the stochastic modelling process. However, existing methods which try to derive procedural rules from given images as proposed by Müller et al. (2007) or Van Gool et al. (2007) still resort to semi-automatic methods. The same holds true for the work of Aliaga et al. (2007). They present an interactive system for both the creation of new buildings in the style of others and the modification of existing buildings. At first, the user manually subdivides a building into its basic external features. This segmentation is then employed to automatically infer a grammar

which captures the repetitive patterns and particularities of the building. Finally, new buildings can be generated in the architectural style defined by the derived grammar. Even though this approach provides individually representative grammars instead of predefined ones, the crucial part of the inference process, the facade interpretation, has to be done manually. In contrast we pursue an approach which runs fully automatically during all processing steps.

The automatic generation of a facade grammar, which is derived from 3D point cloud measurements of a mobile mapping system, are discussed in section 2. As demonstrated in section 3 top-down predictions can be activated and used for the improvement and completion of the reconstruction result that has already been derived from the observed measurements during the bottom-up modelling. Moreover, the facade grammar can be applied to synthesize facades for which no sensor data is available. The discussion of 3D reconstruction results demonstrated in section 4 will conclude the paper.

2. GENERATION OF FACADE GRAMMAR

The automatic generation of a facade grammar based on terrestrial LiDAR data is the core of our facade modelling approach. The first step is a data driven reconstruction process aiming at the detection of geometric facade structures in the observed point clouds. In this regard, a facade defines a planar polygon with holes. Such holes indicate either windows, which will be modelled as indentations, or salient structures such as balconies, oriels or windowsills, which will be attached in the form of protrusions. The result of the data driven facade reconstruction serves as knowledge base for the generation of facade geometries where no sensor data is available. This knowledge, which includes information on dominant or repetitive structures as well as their interrelationships, can be inferred fully automatically and stored as a facade grammar. While data collection will be described as a pre-processing step in section 2.1, the basic concepts of the data driven reconstruction and the subsequent grammar inference will be addressed in section 2.2 and 2.3, respectively.

2.1 Data Collection

The StreetMapper mobile laser scanning system which was used for our experiments collects 3D point clouds at a full 360° field of view by operating four 2D-laser scanners simultaneously. The required direct georeferencing during 3D point cloud collection is realized by the integration of observations from GPS and Inertial Measurement Units (IMU). Figure 1 shows a 3D visualisation of the measured trajectory overlaid to the 3D city model which was also used for the following tests. This 3D city model is maintained by the City Surveying Office of Stuttgart. The roof geometry of the respective buildings was modelled based on photogrammetric stereo measurement while the walls trace back to given building footprints. The trajectory was captured during our tests within an area in the city centre of Stuttgart at a size of 1.5 km x 2km. The respective point clouds were measured at a point spacing of approximately 4cm. Figure 2 depicts a part of the StreetMapper point cloud at the historic Schillerplatz in the pedestrian area of Stuttgart. The observed points are overlaid to the corresponding 3D building models in order to show the quality and amount of detail of the available data. Another measured point cloud overlaid to an existing coarse building model is shown in Figure 3. This example is used in the following to illustrate our bottom-up process for facade reconstruction. Within this process, the geometric information inherent in the available point cloud is exemplarily extracted for the facade marked by the white polygon.

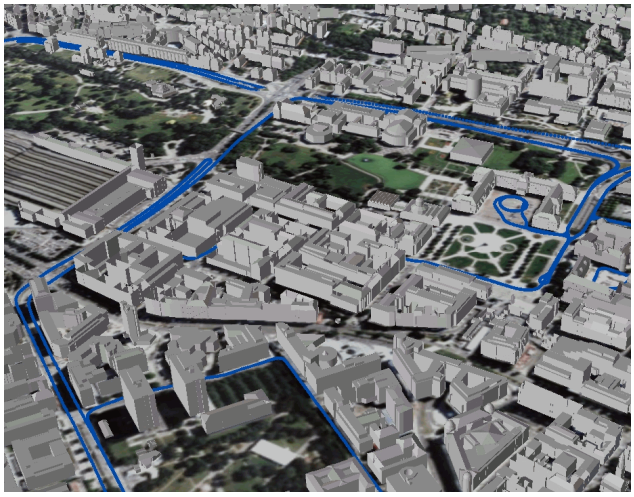


Figure 1. 3D city model with overlaid trajectory from mobile TLS

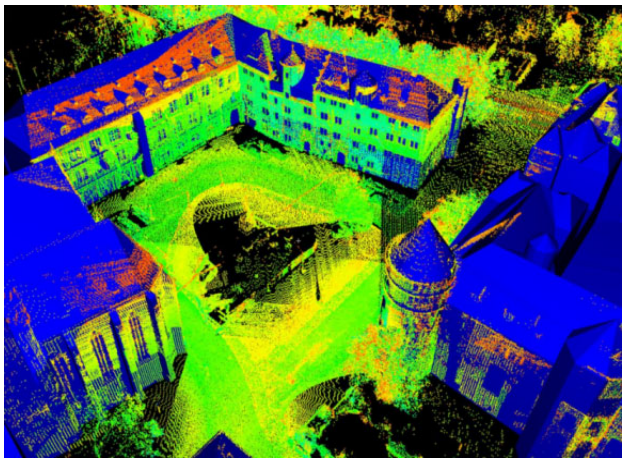


Figure 2. Point cloud from mobile TLS aligned with virtual city model

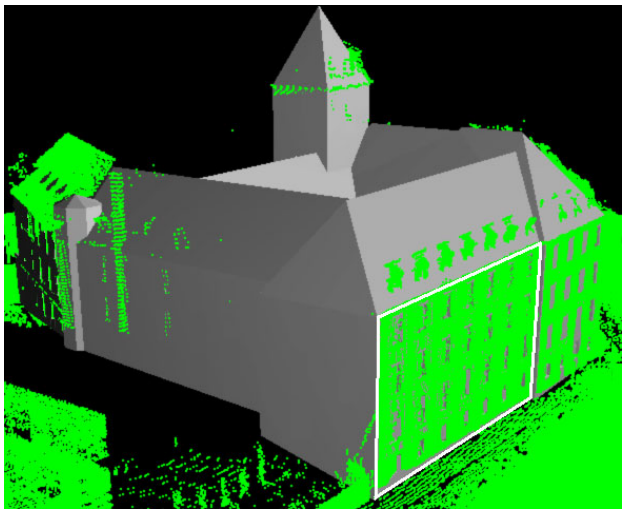


Figure 3. Lindenmuseum, Stuttgart: point cloud from TLS aligned with existing coarse building model

2.2 Data Driven Reconstruction

Frequently, the representation of buildings is based on constructive solid geometry (CSG) or boundary representation (B-Rep). In contrast, we apply a representation of the buildings by cell decomposition (Haala et al., 2006). By these means, problems which can occur during the generation of topologically correct boundary representations can be avoided. Additionally, the implementation of geometric constraints such as meeting surfaces, parallelism and rectangularity is simplified. Due to the applied representation scheme, the idea of our reconstruction algorithm is to segment an existing coarse 3D building object with a flat front face into 3D cells. Each 3D cell represents either a homogeneous part of the facade or a window area. Therefore, they have to be differentiated depending on the availability of measured LiDAR points. After this classification step, window cells are eliminated while the remaining facade cells are glued together to generate the refined 3D building model. These steps are depicted exemplarily within Figure 4 and Figure 5, and will be explained in the following sections. The processing is based on the facade and point cloud marked by the white polygon in Figure 3.

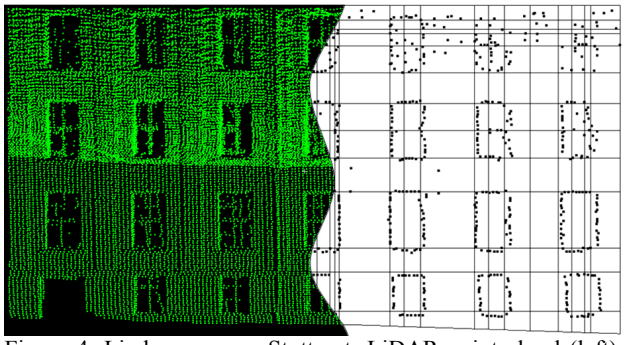


Figure 4. Lindenmuseum, Stuttgart: LiDAR point cloud (left), and detected edge points and window lines (right)

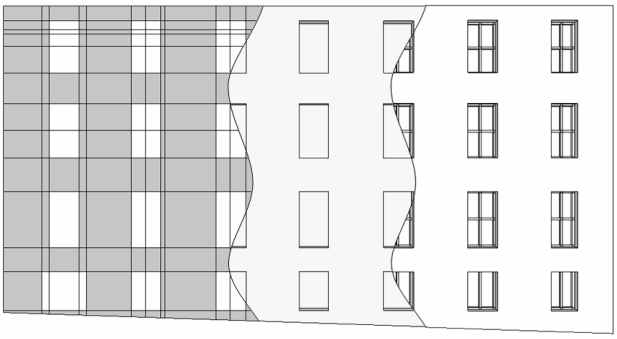


Figure 5. Lindenmuseum, Stuttgart: classified 3D cells (left), 3D facade model (middle), and refined 3D facade model (right)

2.2.1 Point Cloud Segmentation

At glass LiDAR pulses are either reflected or the glass is penetrated. Thus, as it can be seen in Figure 4(left), by laser scanning usually no points are measured in the facade plane at window areas. If only the points are considered that lie on or in front of the facade, the windows will describe areas with no point measurements. These no-data areas can be used for the point cloud segmentation which aims at the detection of window edges. For example, the edge points of a left window border are detected if no neighbour measurements to their right side can be found in a predefined search radius. In a next step, horizontal and vertical lines are estimated from non-isolated edge points. Figure 4(right) shows the extracted edge points at the window borders as well as the derived horizontal and vertical lines. Based on these window lines, planar delimiters can be generated for a subsequent spatial partitioning. Each boundary line defines a partition plane which is perpendicular to the facade. For the determination of the window depth, an additional partition plane can be estimated from the LiDAR points measured at the window crossbars. These points are detected by searching a plane parallel to the facade, which is shifted in its normal direction. The set of partition planes provides the structural information for the cell decomposition process. It is used to intersect the existing building model producing a set of small non-overlapping 3D cells.

2.2.2 Classification and Reconstruction

In order to classify the 3D cells into facade and window cells, a point-availability-map is generated. It is a binary image with low resolution where each pixel defines a grid element on the facade. The optimal grid size is a value a little higher than the point sampling distance on the facade. Grid elements on the facade where LiDAR points are available produce black pixels (facade pixels), while white pixels (non-facade pixels) refer to no-data areas. The classification is implemented by computing the ratio of facade to non-facade pixels for each 3D cell. Cells including more than 70% facade pixels are defined as facade solids, whereas 3D cells with less than 10% facade pixels are assumed to be window solids. While most of the 3D cells can be classified reliably, the result is uncertain especially at window borders or in areas with little point coverage. However, the integration of neighbourhood relationships and constraints concerning the simplicity of the resulting window objects allows for a final classification of such uncertain cells. Figure 5(left) shows the classified 3D cells: facade cells (grey) and window cells (white).

Within a subsequent modelling process, the window cells are cut out from the existing coarse building model. Thus, windows and doors appear as indentations in the building facade which is depicted in Figure 5(middle). Moreover, the reconstruction approach is not limited to indentations. Details can also be added as protrusions to the facade (Becker and Haala, 2007). However, the achievable level of detail for 3D objects that are derived from terrestrial laser scanning depends on the point sampling distance. Small structures are either difficult to detect or even not represented in the data. Nevertheless, by integrating image data with a high resolution in the reconstruction process the amount of detail can be increased (Becker and Haala, 2007). This is exemplarily shown for the reconstruction of window crossbars in Figure 5(right).

2.3 Automatic Inference of Facade Grammar

As it is already visible in Figure 3, the given scan configuration resulted in considerable variations of the available point coverage for the respective building. Thus, the bottom-up facade reconstruction presented in the previous section was realized for a facade, which is relatively well observed. This overall result is now used to infer the facade grammar. Frequently, such formal grammars are applied during object reconstruction to ensure the plausibility and the topological correctness of the reconstructed elements (Müller et al., 2006). In our application, a formal grammar will be used for the generation of facade structure where only partially or no sensor data is available.

In principle, formal grammars provide a vocabulary and a set of production or replacement rules. The vocabulary comprises symbols of various types. The symbols are called non-terminals if they can be replaced by other symbols, and terminals otherwise. The non-terminal symbol which defines the starting point for all replacements is the axiom. The grammar's properties mainly depend on the definition of its production rules. They can be, for example, deterministic or stochastic, parametric and context-sensitive. A common notation for productions which we will refer to in the following sections is given by

$$id: \quad lc < pred > rc : cond \quad \rightarrow \quad succ : prob$$

The production identified by the label *id* specifies the substitution of the predecessor *pred* for the successor *succ*. Since the predecessor considers its left and right context, *lc* and *rc*, the rule gets context-sensitive. If the condition *cond* evaluates to true, the replacement is carried out with the probability *prob*. Based on these definitions and notations, we develop a facade grammar $I^{facade}(N,T,P,\omega)$ which allows us to synthesize new facades of various extents and shapes. The axiom ω refers to the new facade to be modelled and, thus, holds information on the facade polygon. The sets of terminals and non-terminals, *T* and *N*, as well as the production rules *P* are automatically inferred from the reconstructed facade as obtained by the data driven reconstruction process (section 2.2).

2.3.1 Searching for Terminals

In order to yield a meaningful set of terminals for the facade grammar, the building facade is broken down into some set of elementary parts, which are regarded as indivisible and therefore serve as terminals. For this purpose, a spatial partitioning process is applied which segments the facade into floors and each floor into tiles. Tiles are created by splitting the floors along the vertical delimiters of geometries. A geometry describes a basic object on the facade that has been generated during the data driven reconstruction process (section 2.2). It represents either an indentation like a window or a protrusion like a balcony or an oriel. Two main types of tiles can be distinguished: wall tiles, which represent blank wall elements, and geometry tiles, which include structures like windows and doors. All these tiles are used as terminals within our facade grammar. In the remaining sections of the paper, wall tiles will be denoted by the symbols *W* for non-terminals and w_i for terminals. Geometry tiles will be referred to as *G* and g_i in case of non-terminals and terminals, respectively.

2.3.2 Interrelationship between Terminals

Having distinguished elementary parts of the facade we now aim at giving further structure to the perceived basic tiles by grouping them into higher-order structures. This is done fully automatically by identifying hierarchical structures in sequences of discrete symbols. The structural inference reveals hierarchical interrelationships between the symbols in terms of rewrite rules. These rules identify phrases that occur more than once in the string. Thus, redundancy due to repetition can be detected and eliminated. For more information on this process please refer to Becker et al. (2008). As an example, Figure 6a shows a modelled floor. While Figure 6b depicts the corresponding tile string in its original version, the compressed string and the extracted structures are given in Figure 6c. The hierarchical relations between the facade elements can be stored in a parse tree illustrated in Figure 6d.

a)

b) floor 1 → $w_1\ g_1\ w_3\ g_1\ w_1\ g_1\ w_3\ g_1\ w_1\ g_1\ w_3\ g_1\ w_1\ g_1\ w_3\ g_1 \cdots$
$w_2\ g_1\ w_3\ g_1\ w_2\ g_1\ w_3\ g_1\ w_1\ g_1\ w_3\ g_1\ w_1\ g_1\ w_3\ g_1\ w_1\ g_1\ w_3\ g_1\ w_1$

c) floor 1 → $w_1\ S_3\ w_2\ S_1\ w_2\ S_3\ w_1$

$S_1 \rightarrow g_1\ w_3\ g_1$
$S_2 \rightarrow S_1\ w_1\ S_1$
$S_3 \rightarrow S_2\ w_1\ S_2$

d)

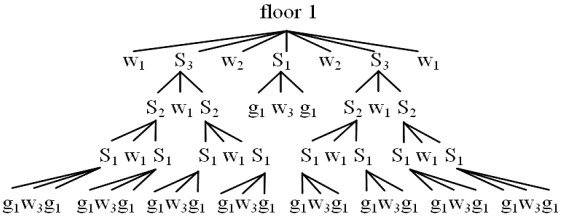


Figure 6. Modelled floor (a), corresponding tile string (b), compressed tile string and extracted structures (c), parse tree (d)

2.3.3 Inference of Production Rules

Based on the sets of terminals $T=\{w_1, w_2, \ldots, g_1, g_2, \ldots\}$ and non-terminals $N=\{W, G, \ldots, S_1, S_2, \ldots\}$, which have been set up previously, the production rules for our facade grammar can be inferred. Following types of production rules are obtained during the inference process:

p_1: $F \rightarrow W+$
p_2: $W : cond \rightarrow W\ G\ W$
p_3: $G : cond \rightarrow S_i : P(\boldsymbol{x}|p_3)$
p_4: $G : cond \rightarrow g_i : P(\boldsymbol{x}|p_4)$
p_5: $lc < W > rc : cond \rightarrow w_i : P(\boldsymbol{x}|p_5)$

The production rules p_1 and p_2 stem from the spatial partitioning of the facade. p_1 corresponds to the horizontal segmentation of the facade into a set of floors. The vertical partitioning into tiles is reflected in rule p_2. A wall tile, which in the first instance can stand for a whole floor, is replaced by the sequence wall tile, geometry tile, wall tile. Each detected structure gives rise to a particular production rule in the form of p_3. This rule type states the substitution of a geometry tile for a structure S_i. In addition, all terminal symbols generate production rules denoted by p_4 and p_5 in the case of geometry terminals g_i and wall terminals w_i, respectively. A more detailed description of all rule types p_i and the probabilities $P(\boldsymbol{x}|p_i)$ assigned to them can be found in Becker et al. (2008).

3. APPLICATION OF FACADE GRAMMAR

Our facade grammar derived in the previous section implies information on the architectural configuration of the observed facade concerning its basic facade elements and their interrelationships. Based on this knowledge facade hypotheses can be generated as described in section 3.1. Section 3.2 presents different application scenarios. Facades and building parts which are covered by noisy or incomplete sensor data are usually subject to inaccurate and false reconstructions which are due to problems of the data driven reconstruction process. For such regions possible facade geometry can be proposed in order to improve and complete facade structures. Furthermore, the production process can also be used to synthesize totally unobserved building objects.

3.1 Production of Facade Hypothesis

The production process starts with an arbitrary facade, called the axiom, and proceeds as follows: (1) Select a non-terminal in the current string, (2) choose a production rule with this non-terminal as predecessor, (3) replace the non-terminal with the rule's successor, (4) terminate the production process if all non-terminals are substituted, otherwise continue with step (1). The geometrical result of the production process depends on the order in which the non-terminals are selected. Usually, best results are obtained when facade structures which are likely to appear in the middle of the facade are placed first, and the remaining spaces to the left and the right side are filled afterwards. As it is illustrated in Figure 7, the non-terminal selection refers to this principle. For clearness, we here assume a facade with only one floor. In each step, the non-terminal selected for the next substitution is marked in red.

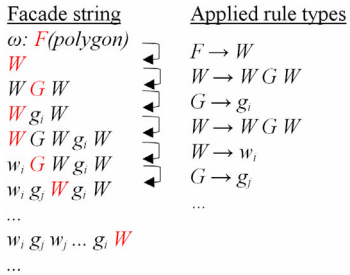


Figure 7. Non-terminal selection

As long as the facade string consists of only one symbol, the non-terminal selection is trivial. In the third line, substitution starts with the non-terminal G in the middle of the string. According to this replacement, the chosen geometry tile g_i will be placed about in the middle of the facade floor. The following replacements are taken from the left to the right of the string. When there is only one non-terminal left on the right end of the string (see the last line in Figure 7), the left part of the facade floor is completely filled with a sequence of wall and geometry tiles. At this stage, symmetry can be enforced by substituting the remaining non-terminal W by a mirrored version of the left terminal string. If no symmetry is required, the replacement can be continued as described before. During the production, non-terminals are successively rewritten by the application of appropriate production rules. When more than one production rule is possible for the replacement of the current non-terminal, the rule with the highest probability value is chosen. As soon as the facade string contains only terminals, the production is completed and the string can be transferred into a 3D representation.

3.2 Application Scenarios

Within the production process, the grammar is applied to generate hypotheses about possible positions of each geometry tile and thereby synthesize facade geometry for given coarse building models. This process can for example be used to generate facade structure at areas, where sensor data is only available at limited quality. Such a scenario is depicted exemplarily in Figure 8, which shows a StreetMapper point cloud for an exemplary facade acquired during two epochs. The colours encode the different scanners mounted on the StreetMapper. Points that stem from the upward facing laser scanner are marked in yellow; points that are measured by the side facing scanners are blue (right scanner) and red (left scanner).

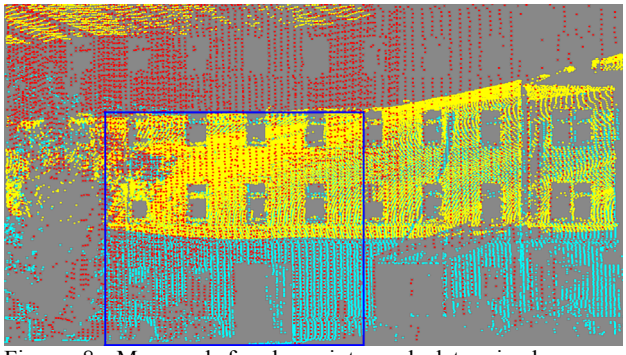


Figure 8. Measured facade points and determined convex 'dense area' (blue rectangle)

As it is visible in Figure 8, the point sampling distance varies strongly due to occlusions and oblique scanning views to the upper part of the building. For this reason, facades may contain areas where no or only little sensor data is available. In such regions, an accurate extraction of windows and doors cannot be guaranteed anymore. Nevertheless, a grammar based facade completion allows for meaningful reconstructions even in those areas. The main idea is to derive the facade grammar solely from facade parts for which dense sensor data and thus accurate window and door reconstructions are available. The detection of such 'dense areas' is based on a heuristic approach evaluating the sampling distances of the points lying on the facade surface. In Figure 8 the extracted convex dense area is marked by a blue rectangle. Since the inference process is restricted to this dense area, a facade grammar of good quality can be provided, which is then used to synthesize the remaining facade regions during the production step.

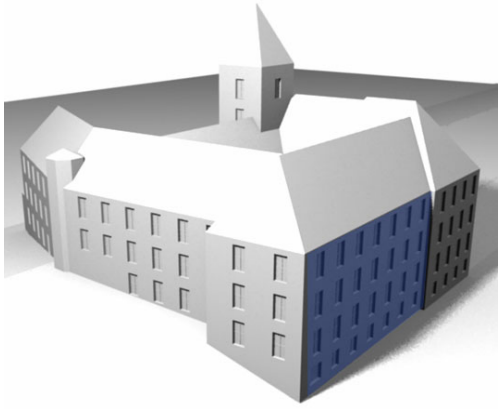


Figure 9. Facade reconstruction for the "Lindenmuseum"

Figure 10. Facade geometry synthesized from grammar library

As an example, this process is depicted in Figure 9 for Stuttgart's Lindenmuseum which has already been illustrated in Figure 3. There, the original coarse model is shown in combination with the overlaid 3D point cloud whereas Figure 9 demonstrates the reconstructed facade geometry. The blue shaded region corresponds to the white polygon in Figure 3 and indicates the facade geometry that has been generated during the data driven reconstruction process. All remaining building parts are modelled based on the grammar inferred from the marked region.

While in that example the grammar is applied for the completion of facade structure, it can also be used as a "library" to generate building facades for objects, where no measurement is available at all. This step is demonstrated in Figure 10 where facade geometry was synthesized for a number of residential houses. Since these building were not covered by any sensor data at all, a range of grammars was derived in advance from a few buildings in the near environment. Similarly, the applicability of our facade grammars to a larger scene using grammars which represent compatible architectural styles is shown in Figure 11.

4. DISCUSSION

Within the paper an automatic approach for the geometric modelling of 3D building facades was proposed. Based on observed 3D point clouds from a mobile mapping system, grammar rules are extracted, which can then be used to generate synthetic facade structures for unobserved building parts. Despite the good geometric accuracy which is feasible for terrestrial point clouds such data frequently suffer from the unavailability of measurements for hidden building parts. This problem is solved by extracting the grammar from observed street-facing facades and then applying it for the improvement and completion of remaining facade structure in the style of the respective building. Moreover, knowledge propagation is not restricted to facades of one single building. Based on a small set of facade grammars derived from just a few observed buildings, facade reconstruction is also possible for whole districts featuring uniform architectural styles. Due to these reasons the proposed algorithm is very flexible towards different data quality and incomplete sensor data.

5. REFERENCES

Aliaga, D., Rosen, P., Bekins, D., 2007. Style Grammars for Interactive Visualization of Architecture. IEEE TVCG 13 (4).

Becker, S., Haala, N., 2007. Refinement of Building Facades by Integrated Processing of LIDAR and Image Data. IAPRS & SIS Vol. 36 (3/W49A), pp. 7-12.

Becker, S., Haala, N., Fritsch, D., 2008. Combined Knowledge Propagation for Facade Reconstruction. IAPRS & SIS Vol. 37 (B5), pp. 1682-1750.

Haala, N., Becker, S., Kada, M., 2006. Cell Decomposition for the Generation of Building Models at Multiple Scales. IAPRS Vol. 36 (3), pp. 19-24.

Haala, N., Peter, M., Kremer, J., Hunter, G., 2008. Mobile LiDAR Mapping for 3D Point Cloud Collection in Urban Areas - a Performance Test. IAPRS, Vol. 37, (B5), pp. 1119f.

Kremer, J., Hunter, G., 2007. Performance of the StreetMapper Mobile LIDAR Mapping System in "Real World" Projects. Photogrammetric Week '07, pp. 215-225.

Müller, P., Wonka, P., Haegler, S., Ulmer, A., Van Gool, L., 2006. Procedural Modeling of Buildings. ACM Transactions on Graphics (TOG) 25 (3), pp 614-623.

Müller, P., Zeng, G., Wonka, P., Van Gool, L., 2007. Image-based Procedural Modeling of Facades. ACM Transactions on Graphics (TOG) 26 (3), article 85, 9 pages.

Prusinkiewicz, P., Lindenmayer, A., 1990. The algorithmic beauty of plants. New York, NY: Springer.

Ripperda, N., 2008. Determination of Facade Attributes for Facade Reconstruction. IAPRS & SIS Vol. 37 (B3a), 6 pages.

Van Gool, L., Zeng, G., Van den Borre, F., Müller, P., 2007. Towards mass-produced building models. IAPRS & SIS, Vol. 36 (3/W49A), pp. 209–220.

Wonka, P., Wimmer, M., Sillion, F., Ribarsky, W., 2003. Instant architecture. ACM TOG 22 (3), pp. 669–677.

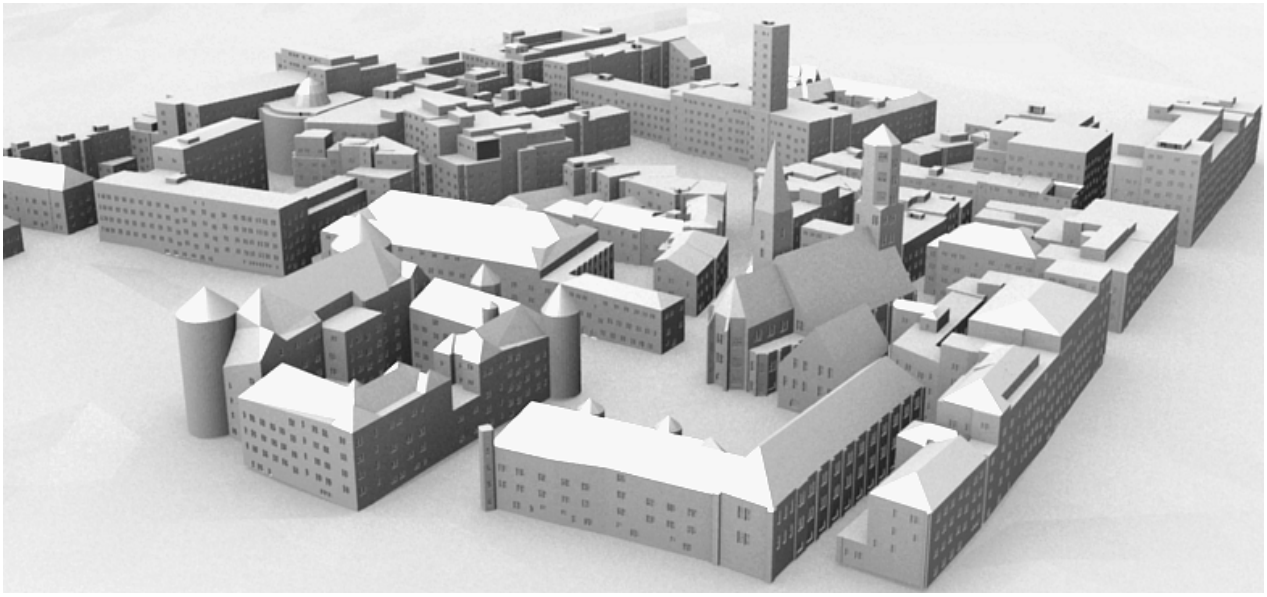


Figure 11. Facade geometry for larger area

Author Index