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.

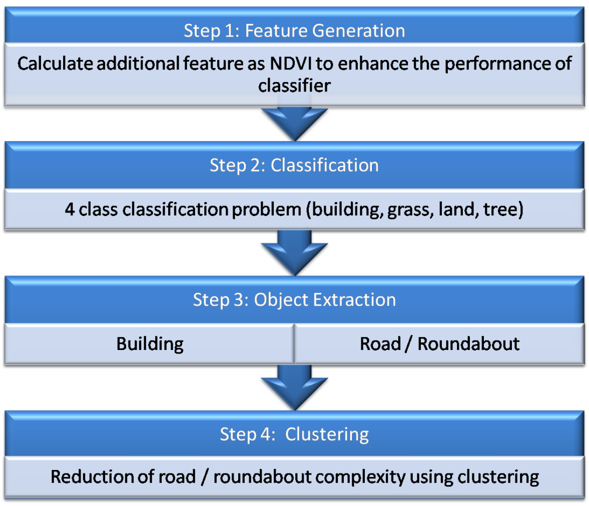
Step 1: Feature Generation
Calculate additional feature as NDVI to enhance the performance of classifier
Step 2: Classification
4 class classification problem (building, grass, land, tree)
Step 3: Object Extraction
Building
Road / Roundabout
Step 4: Clustering
Reduction of road / roundabout complexity using clustering

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.

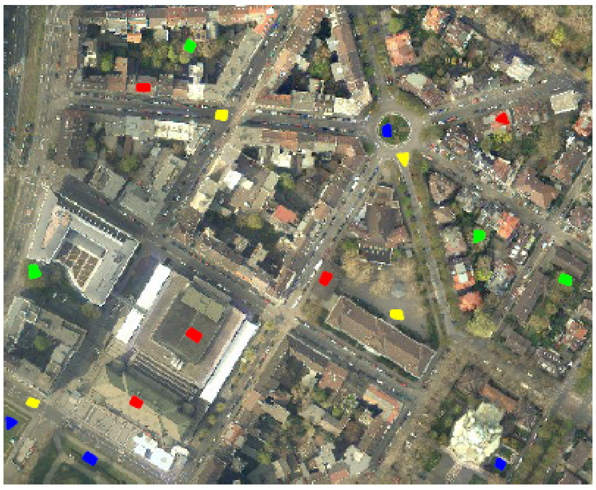

Figure 2: Data and Training set. Red stands for building, yellow for land, blue for grass and green for tree

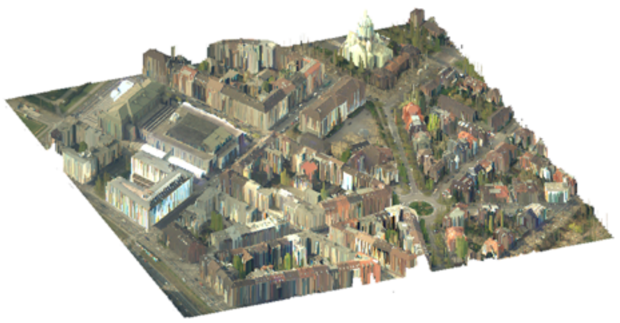

Figure 3: A 3D view of dataset; height of objects are obtained using the first pulse laser range data

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.

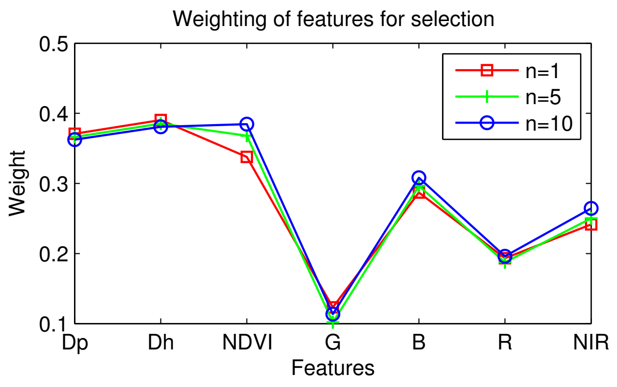

Figure 4: Results of ReliefF algorithm applied to the set of seven features; the n parameter represents the number of nearest instances from each class.

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.

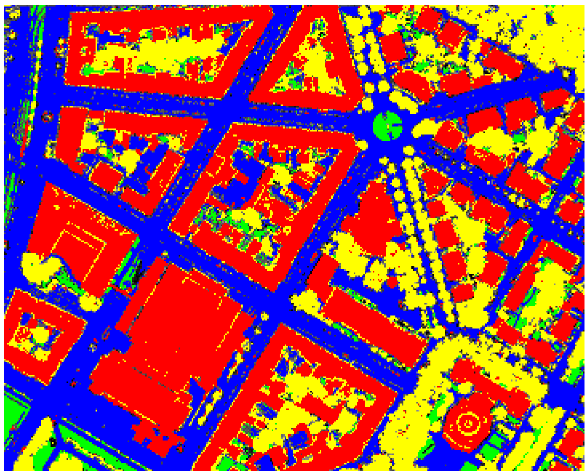

Figure 5: Results of classification using AdaBoost and the training set of 32 polygons; red stands for building, yellow for tree, blue for land and green for grass

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

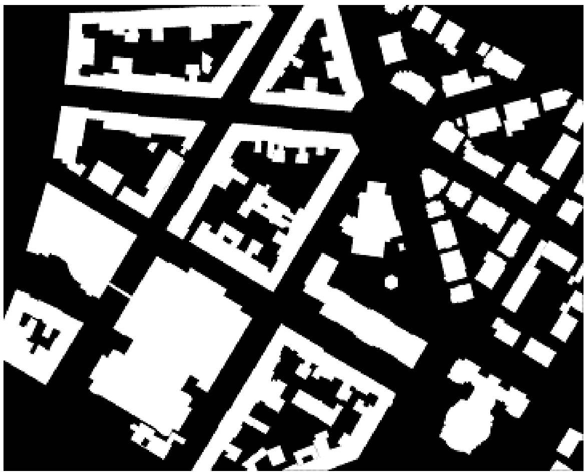

Figure 6: Ground truth used to evaluate the classification results; white pixels are buildings, blacks one are remaining objects

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.

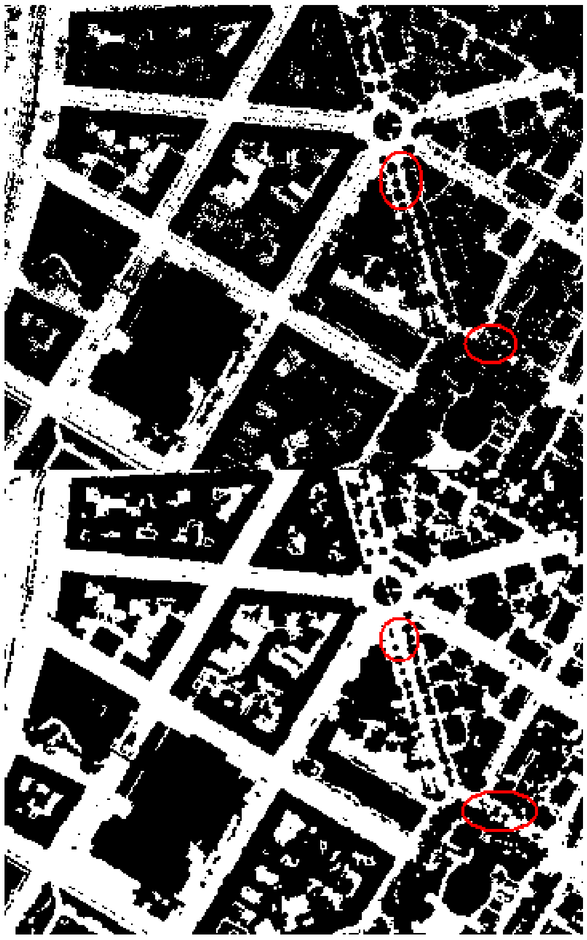

Figure 7: Filtering. In the top image white pixels are classified as land; classification is noisy due to the presence of small objects as vehicles; in the bottom, the non-linear filter allows to reduce significantly the effect of noise and occlusions

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.

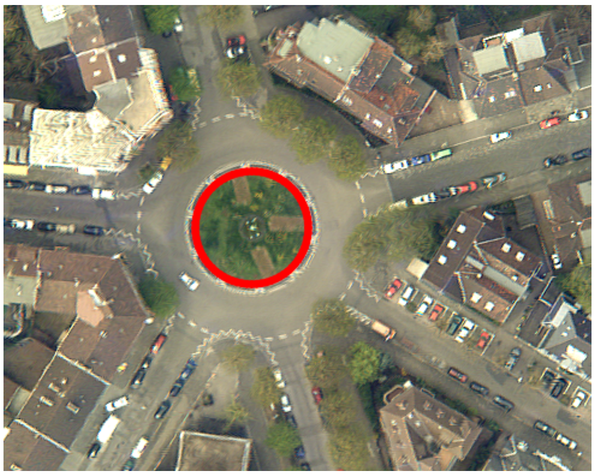

Figure 8: Hough transform applied to Mannheim dataset to find circular shapes as roundabouts

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
 1: $S$ vector of extracted segments
 2: $s$ vector of candidate pixels belonging to a segment
 3: $p$ vector of aligned pixels
 4: for $j = 0$ to $j < height$ do
 5:   for $i = 0$ to $i < width$ do
 6:     for $\theta = -\pi/2$ to $\pi/2$ do
 7:       $p \leftarrow calculate\_segment\_points(i, j, \theta)$
 8:       $start \leftarrow 0$
 9:       $s.clear$
10:       for $k = 0$ to $k < p.size$ do
11:         $n = count\_spurious\_pixels(s, start, x)$
12:         if $n > T2 \vee i == (width - 1) \vee j == (height - 1)$ then
13:           if $p.length > T1$ then
14:             S.add(s)
15:             s.clear
16:             $start \leftarrow k + 1$
17:           else
18:             $s.add(p[k])$
19:           end if
20:         end if
21:         $k \leftarrow k + 1$
22:       end for
23:       $\theta \leftarrow \theta + 1$
24:     end for
25:     $i \leftarrow i + 1$
26:   end for
27:   $j \leftarrow j + 1$
28: end for
```

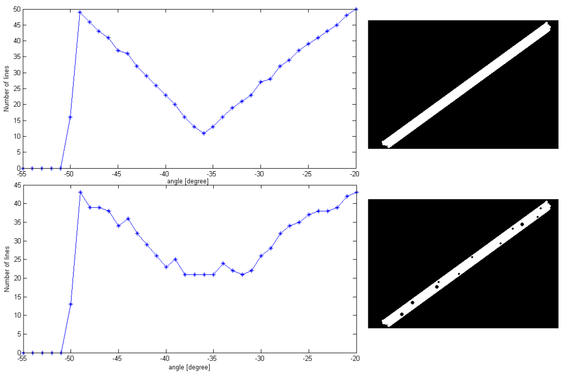

Figure 9: Segment extraction. Top image represent an ideal segment extraction while in the bottom it is tested a noisy image

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).

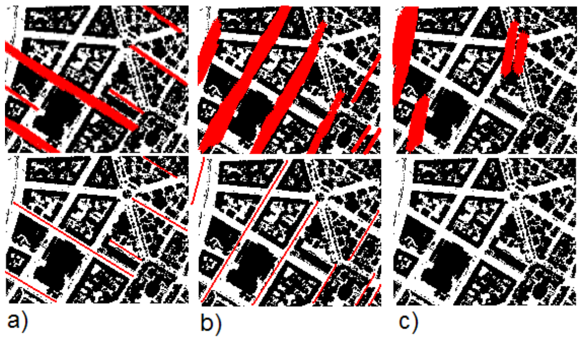

Figure 10: Road extraction for three different angles; segments are the thick red lines (bottom), while raw ones are shown in top

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.

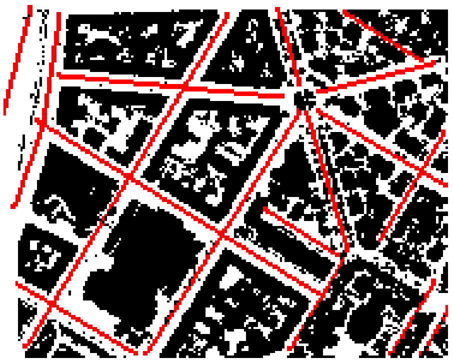

Figure 11: Road graph for Mannheim data-set

5 CONCLUSIONS AND FUTURE WORK

In this paper, we presented a complete methodology to solve the problem of automatic extraction of urban objects from multi-source 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.