AUTOMATIC BUILDING DETECTION FROM LIDAR POINT CLOUD DATA

Nima Ekhtari, M.R. Sahebi, M.J. Valadan Zoej, A. Mohammadzadeh

Faculty of Geodesy & Geomatics Engineering, K. N. Toosi University of Technology, P O Box 15875-4416, Tehran, Iran - nima_e1983@yahoo.com, sahebi@kntu.ac.ir, valadanzouj@kntu.ac.ir, ali_mohammadzadeh2002@yahoo.com

Commission, WG IV/3

KEY WORDS: Building Detection, LIDAR, DSM, DTM, Normalized DSM

ABSTRACT:

This paper proposes an automatic system which detects buildings in urban and rural areas by the use of first pulse return and last pulse return LIDAR data. Initially both first and last pulse return points are interpolated to raster images. This results to two Digital Surface Models (i.e. DSM) and a differential DSM (i.e. DDSM) is computed by them. Then using a height criterion, rough and smooth regions of the DDSM are found. Then last pulse points lying inside smooth regions are filtered using a simplified Sohn filtering method to find the so called on-terrain points by which the Digital Terrain Model (i.e. DTM) is generated. The normalized DSM (i.e. nDSM) is calculated using first pulse-derived DSM and the calculated DTM. Afterwards two separated classifications are applied on the nDSM. The final results of classifications are a set of nDSM pixels belonging to building roofs. The accuracy of the proposed algorithm is evaluated using some metrics and has proved an overall accuracy of 95.1% and a correctness equal to 98.3% and a completeness factor equal to 89.5% which show the level of the efficiency and accuracy of the system.

1. INTRODUCTION

Nowadays there is an increasing demand for 3D urban models produced from Earth Observation data. Such a model contains all buildings of a city superposed on an accurate DTM. 3D urban models are being widely used in the development of 3D GIS databases which has many applications in utility services, traffic management, air pollution control, etc.

A 3D City Modeling procedure consists of three phases that is building detection, building extraction, and building reconstruction. The purpose of first phase is to detect (the pixels belonging to) some regions representing buildings. The subject of second phase is to compute the geometry of polygons which best fit the detected pixels. The third phase aims to compute and fit the best planar roof type for buildings.

Among all available methods to extract building models of a city, those who use integrated data sources appear more successful since the weakness points of either data sources can be compensated by the other one. Many researches have been done on the combination of high-resolution imagery and LIDAR data to detect and extract buildings (Sohn and Dowman 2007; Schenk and Csatho 2002; Rottensteiner et al. 2005; Guo and Yasuoka 2002).

Considering the capability of dense LIDAR data, there is no necessity to involving any aerial images in the building detection task. Even single-source data systems can work faster and more automatically. Many researches have shown the capability of LIDAR data in detection and extraction of the buildings (Vosselman 1999; Maas and Vosselman 1999; Zhang et. al 2006).

Two ways are often utilized to identify building measurements from LIDAR data. One is to separate the ground, buildings, trees, and other measurements from LIDAR data simultaneously. The more popular way is to separate the ground from non-ground LIDAR measurements first and then identify the building points from non-ground measurements [Zhang et. al 2006]. The proposed algorithm here is also among the latter way.

Although there have been some automatic and semi-automatic methods of building modeling proposed by many researchers, all the three phases still can be studied and developed more. The fact that the accuracy of building detection phase has a dominant direct effect on the buildings extraction task, suggests that there is a need for more accurately detected building pixels. Because the geometry of buildings are extracted wherever building pixels are detected. This paper develops a building detection system.

Our building detection algorithm (system) is therefore a single-source data system, since it only utilizes LIDAR data to detect buildings. Our system generates some Digital Elevation Models by the interpolation of LIDAR points. The result of our system is presented in a raster format. In other words our algorithm uses vector data (3D coordinated points) as inputs and gives the information in raster format (Building pixels with their elevation).

2. STUDY AREA

The LIDAR point cloud data used to evaluate our system comprises of two recorded laser pulse returns; First pulse and Last pulse return points. FP (i.e. First Pulse return) points are those recorded from the first reflection of the laser pulse. As a result they might belong to the edges or surfaces of objects on the terrain rather than the ground beneath them. While the LP (i.e. Last Pulse return) points are more likely to belong to the terrain surface, especially for points of vegetation-covered regions and those near walls of buildings. That's why we prefer LP points to create DTM and FP points to create DSM.

The dataset used to evaluate the accuracy of our algorithm contains both the FP and LP data. The points of either have a

1.2 meters across-track and 10 centimeters along-track spacing. The dataset is provided by the ISPRS Commission III Working group8 official web-site, and is available on-line at: http://isprs.ign.fr/packages/zone3/package3_en.htm

An aerial image with 25 centimeters ground pixel size is also provided from the scene which is shown in Fig. 1. This image can be useful for visual comparisons.

Figure 1 – Aerial image of the study area

3. IMPLEMENTATION

The building detection system starts with a classification process which makes use of both FP and LP points. This classification divides the LIDAR points into "Rough" and "Smooth" classes. Of course as will be described later, many points within dense trees will be misclassified in "Smooth" point class. Then a simplified version of the so-called Sohn filter is used to extract on-terrain points from the points of "Smooth" class and the DTM is generated using these points. The normalized DSM can be computed by the DSM and DTM. Then a thresholding separates high-rise pixels from the nDSM. These pixels may belong either to building roofs or to dense vegetation covers. Then a slope thresholding applied on the slope map of the nDSM arranges the pixels of nDSM into either of the two classes of "Severely" and "Slightly" variable slope pixels. Finally building pixels are detected among the members of "Slightly Variable" class which simultaneously belong to the "High-rise" class. The whole procedure is described in details in the following subsections.

3.1. DSM roughness analysis

As mentioned before, in order to reduce the amount of calculations in the Sohn filter, our system tends to find the points belonging to "Rough" areas and filters them out. Such points in both FP and LP data have different heights due to the canopy penetration capability of laser pulse. So a simple way to detect these points is the subtraction of the heights of all points in last pulse return from corresponding points in first pulse return. The only problem is that the height differences from the first and last returns do not work for areas covered by dense trees where laser pulses cannot penetrate [Zhang et. al 2006]. This will cause many points of dense vegetated areas to remain among "Smooth" points.

Often the points of first and last returns of laser do not necessarily have the same exact planar coordinates since the scan angle is not perpendicular to terrain. This case happens predominantly wherever the elevation changes abruptly like vegetated areas and near the walls of buildings. To tackle this problem we generate two Digital Surface Models (i.e. DSM) by interpolating FP and LP points individually. The height difference of corresponding pixels in these two models is stored in an image called the differential DSM (i.e. DDSM) image. The value of the pixels of DDSM is more wherever the pixels belong to vegetations or walls.

A threshold equal to 15 centimeters is set to discriminate vegetation from other covers in the DDSM. Pixels with values more than the threshold are classified as "Rough" pixels and the rest of pixels will be assigned the "Smooth" label. The pixels of "Smooth" class then make a mask image (Fig. 2). Every LIDAR point which lies inside the mask should contribute in the generation of the Digital Terrain Model and hence these points are stored in an individual file labeled "Smooth points". Fig. 2 shows the classified DDSM on which the pixels of "Rough" class are assigned a green color, while yellow pixels represent the "Smooth" class.

Figure 2 – The result of the classification of DDSM pixels into "Smooth" (light tone) and "Rough" (dark tone)

3. 2. Filtering the LP data

In order to generate the Digital Terrain Model from LIDAR data, a filtering process is implemented on the LP data. The result of this filtering is a set of points which lie on the terrain. A filtering method called the "Sohn filter" (G.Sohn, I.Dowman 2002) -also called "Progressive TIN densification/ Regularization method" by some authors- is the basis of our filtering step. Their algorithm is based on a two-step progressive densification of a TIN; the Points in the TIN at the end of the densification are accepted as a representation of the bare earth, and the rest as object [Sithole 2005]. We have done our filtering based on a simplified version of their algorithm. The first step of densification in our filtering is somehow the same as Sohn's. The only difference is that we select more than four points as initial on-terrain points. But we have made some simplifications in the second step, where we have ignored the MDL (i.e. Minimum Description Length) criterion. Our study area is almost a flat, smoothly sloped area with a few flat roofed buildings. Since there is no dominant topographic influence in the scene, investigating the MDL criterion is not a necessary task. That's why we have made the aforementioned simplification.

All the points inside the "Smooth points" file are the inputs to the filtering step. A set of initial on-terrain points including four points covering the study area, and a few points (three points in this case) at the middle of the scene are selected and the triangulation is triggered by them. The selection of these points is not a difficult task since they are members of the "Smooth" class of the DSM. Then lowest point in each triangle is found and added to the on-terrain points group and the triangulation is repeated again. This procedure is iterated until there is no point below any triangle. All the points of the last TIN are assigned an on-terrain label.

The second step of densification starts with the final TIN made in the last step. A buffering space with a distance of 50 centimeters is defined above each triangle. All of the points in the "Smooth points" file except for those used in the TIN are examined. Every point within the buffer is assigned an on-

terrain label. This procedure is not iterative. Finally the on-terrain points are interpolated to a raster image using the Natural Neighbor interpolation method to produce the Digital Terrain Model (i.e. DTM). Every pixel of DTM has a real coordinate in the object space and its value is proportional to the elevation of the terrain at that position. Fig3 shows the calculated DTM which is classified into 5 classes to show the slope aspect of the study area. The maximum and minimum elevations of DTM pixels are 25.94 and 23.17 meters respectively.

Figure 3 – Calculated Digital Terrain Model (darker tones symbolize the higher pixels)

3.3. Elevation analysis of nDSM

Since buildings are highly elevated objects in a scene our system looks for high-rise objects in the study area in this step. A high-elevated pixel on a DSM may belong to any objects as well as a high region of terrain. To eliminate the effect of topography from a DEM, one should normalize the model. A surface model is normalized by the use of the corresponding terrain model. To normalize the DSM, we subtract the value of each pixel of DTM from the value of the corresponding pixel in DSM. The result is a normalized model called nDSM (i.e. normalized DSM).

As shown in Fig. 4 on-terrain pixels have values (i.e. elevations) less than 10 centimeters. A 10 centimeters threshold classifies these pixels. This supports the fact that bare-earth segments of the scene have the same elevations in both DSM and DTM. A "Terrain" label is assigned to these pixels. The next pace is finding the locally highest pixels of the nDSM. These pixels represent objects which lie on the terrain. A threshold equal to 3 meters derives the high-rise objects of the scene. As shown in Fig. 4 this threshold has classified the rest of nDSM pixels into two other classes which are "Low- rise" and "high-rise" objects classes.

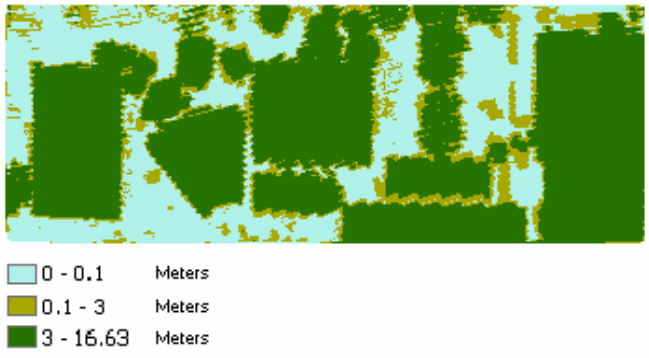

Figure 4 – Classified normalized DSM (height thresholding)

So we classified all nDSM pixels into three classes; "Terrain", "Low-rise", and "High-rise". The pixels of "High-rise" class are the input to the last step of our building detection procedure.

3.4. Roughness analysis of nDSM

It is conspicuous that not all "High-rise" pixels belong to buildings. As explained earlier (section 3.1) some dense vegetation regions weren't detected as "Rough" regions. So the presence of tree pixels in "High-rise" pixels is also expectable as shown in Fig. 5. The goal of this step is the detection of building pixels among the pixels of "High-rise" class.

To fulfill this task we have used a simple concept of surface roughness that is the computation of the slope map of the nDSM image. The main motivation is that the slope of the roofs of buildings doesn't often change abruptly. In addition, our study area contains a few flat-roofed buildings. Consequently we computed the slope map of the nDSM using the ESRI ArcGIS9.2 software. The amount of slope for each pixel is computed by this software using this formula [Burrough 1998]:

$$\text{slope\_radians} = \text{ATAN}\left(\sqrt{ \left([dz/dx]^2 + [dz/dy]^2 \right) } \right)$$

Where **[dz/dx]** and **[dz/dy]** are the rate of height change in X and Y directions respectively. These rates are computed for a 3 * 3 cell neighborhood around every pixel.

So the slope map of the nDSM image is computed and generated. Then a quick trial and error method leads a human operator to define an appropriate threshold by which the slope map of nDSM can be divided into two classes; "Severely variable" and "Slightly variable" regions. The members of the former class are those pixels representing high-rise vegetations and walls of buildings, and the members of the latter class are the representatives of building roofs and relatively flat areas on the terrain. Fig. 5 shows the results of the thresholding the slope map of nDSM where the threshold is set to 5% of the slope range of the slope map.

Figure 5 - Classified normalized DSM (slope thresholding) – Dark and light tones represent "Severely" and "Slightly" variable areas respectively

3.5. Detection of building pixels

So far we have obtained an nDSM image with three classes; "Terrain", "Low-rise", and "High-rise" regions, as well as a slope map of nDSM with two classes; "Severely variable" and "Slightly variable" regions.

As stated in section 3.3 buildings are highly elevated objects in a scene. In section 3.4 we also mentioned that the most of the buildings have smoothly sloped roofs. These facts suggest that the building pixels are those pixels of "high-rise" regions which their counterpart in the slope map belongs to the "Slightly variable" regions. In other words each pixel which belongs to both of these classes is a building point. This way our system detects building pixels and labels them as "Building". All the remaining pixels of the scene are classified as "Non-building".

The resulting image is shown in Fig. 6(a). It is obvious that still some dense, high-rise vegetation are misclassified as building.

The main reason is that these high-rise segments of study area are covered densely by vegetations so that no laser penetration is possible there. This causes them to be classified in both "High-rise" and "Slightly variable" regions, and consequently to be misclassified as "Building". In order to eliminate these erroneous pixels from "Building" class, an area threshold equal to 70 square meters is defined and applied to the resulting image. The final results are shown in Fig. 6(b). As can be seen all the remaining polygons are parts of the roofs of buildings.

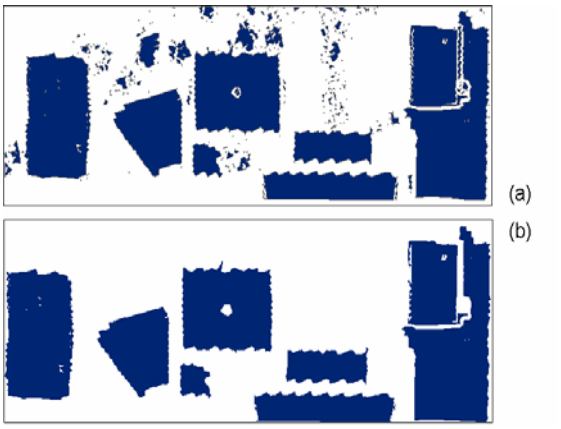

Figure 6 - (a) detected building pixels -including high-rise, dense vegetation- (b) Building detection final results

At the middle of some building roofs in Fig. 6(b) there are some pixels classified as non-building while they also should belong to the building class. This happens wherever a high-rise feature exists on the roof. In the case of the large building at the east of the scene, the height difference on the roof changes so abruptly that our system fails to detect the whole building. As a result, our system has detected two individual buildings there.

4. ACCURACY ASSESSMENT

The final output of our system is a raster image with pixels classified into two classes; "Building" and "Non-building" pixels. To assess the accuracy of the detected buildings, we have compared our results of building detection with the reference map provided by the data provider. Due to some unknown reasons one of the buildings of the scene is not defined in the reference map. So we excluded the corresponding detected pixels from our results. Fig. 7 shows our results and the reference map.

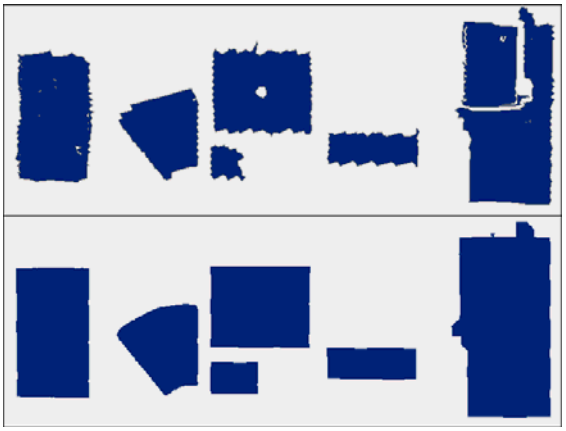

Figure 7 – Building detection results versus Ground truth

The comparison of the resulting image with the reference map is done by the calculation of the Confusion Matrix using the RSI ENVI4.2 software. Table (1) shows the computed confusion matrix.

Confusion matrix	**Reference Map**		
Our Results	Building	Non-Building	Total
Building	**82185 (TP)**	**1380 (FP)**	**83565**
Non-Building	**9596 (FN)**	**133783 (TN)**	**143379**
Total	**91781**	**135163**	**226944**

Table (1) The pixel-by-pixel comparison results for our system.

In Table (1), TP (i.e. True Positive) shows the number of pixels which have a "Building" label in both datasets. Similarly TN (i.e. True negative) equals to the number of pixels having "Non-building" labels in both compared datasets. The definition of FN and FP numbers is straightforward.

The evaluated data in Table (1) are the results of a pixel-by pixel comparison between our results and the reference map. The two following objective metrics [Lee et. al 2003] are employed by some authors (Sohn and Dowman 2007) in order to provide a quantitative assessment of our building detection system.

Completeness = 100 * (TP / TP + FN)
Correctness = 100 * (TP / TP + FP)

The evaluated amount of Completeness metric for our results equals to 89.5% which shows the building detection percentage [Sohn and Dowman 2007]. And the amount of Correctness metric for our results equals to 98.3% which shows this percentage of the "Building" detected pixels belong to buildings indeed.
The commission and omission errors are also evaluated in percentages and listed in Table (2). Errors of commission represent pixels that belong to another class that are labeled as belonging to the class of interest. Errors of omission represent pixels that belong to the ground truth class but the classification technique has failed to classify them into the proper class.
For instance, the amount of 1.65 for Commission error of "Building" class states that 1.65% of pixels in "Building" class do not really belong to building roofs. The Omission error of "Building" class in Table (2) shows that 10.46% of building-roof pixels have been misclassified by our system as "Non-building". It's obvious that Commission and Omission errors for "Building" class are complementary amounts to the aforementioned Correctness and Completeness metrics.

Class	**Error**	**Commission**	**Omission**
Building		**1.65**	**10.46**
Non-building		**6.69**	**1.02**

Table (2) The evaluated Commission and Omission errors for our results in percentages

The Overall accuracy is another metric which evaluates the accuracy of any classification process. This metric can be evaluated using the following formula:

Overall Accuracy = (TP + TN) / Total number of pixels

The Overall accuracy of our building detection system equals to 95.1% which shows the percentage of correctly classified pixels.

5. CONCLUSIONS AND FURTHER REASEARCH

This paper presented an automatic building detection system by the use of LIDAR point cloud data provided in two individual files as first pulse and last pulse returns of laser pulse. Our system is comprised of five steps explained successively in section 3. In the resulting image, pixels are assigned either "Building" or "Non-building" labels.

The results of the pixel-by-pixel comparison method used here proved that our building detection system has made some improvements in the detection task in comparison with some previous works. In addition to the Completeness and Correctness metrics, the evaluated 95.1% Overall accuracy of our system proves its efficiency and relatively high accuracy. We hope to make a clear comparison between our method and the existing ones in future works to claim the capabilities of our method.

The study area contains a few flat-roofed, distanced buildings with some high-rise vegetation between them. So the accuracy of the system is not tested for any other scenes including complex scenes or dense urban areas. It is also recommended that the main version of Sohn filtering be used for more complex scenes and areas with significant topographic change effects.

Different datasets including buildings with various roof types are recommended to be included for the assessment of the system, or any further corrections to the whole algorithm.

REFERENCES

Burrough, P. A. and McDonell, R.A., 1998. *Principles of Geographical Information Systems* (Oxford University Press, New York), p. 190.

Guo, T., Yasuoka, Y., 2002. Snake-based approach for building extraction from high-resolution satellite images and height data in urban areas. Proceedings of the 23rd Asian Conference on Remote Sensing, November 25–29, Kathmandu. 7 pp., on CD-ROM.

Keqi Zhang, Jianhua Yan, and Shu-Ching Chen. Automatic Construction of Building Footprints From Airborne LIDAR Data. IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING, VOL. 44, NO. 9, SEPTEMBER 2006

Lee, D.S., Shan, J., Bethel, J.S., 2003. Class-guided building extraction from IKONOS imagery. Photogrammetric Engineering and Remote Sensing 69 (2), 143–150.

Maas, H.-G., Vosselman, G., 1999. Two algorithms for extracting building models from raw laser altimetry data. ISPRS Journal of Photogrammetry and Remote Sensing 54 (2–3), 153–163.

Rottensteiner, F., Trinder, J., Clode, S., Kubik, K., 2005. Using the Dempster–Shafer method for the fusion of LIDAR data and multi-spectral images for building detection. Information Fusion, Volume 6, Issue 4, December 2005, pp 283-300

Schenk, T., Csatho, B., 2002. Fusion of LiDAR data and aerial imagery for a more complete surface description. International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences 34 (Part 3), 310–317

Sithole, G., Segmentation and Classification of Airborne Laser Scanner Data (PhD Thesis). Publications on Geodesy 59 **NCG (**Netherlands Geodetic Commission) Delft, May 2005

Sohn, G., Dowman, I.J., 2002. Terrain surface reconstruction by the use of tetrahedron model with the MDL criterion. International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences 34 (Part 3), 336–344.

Sohn, G., Dowman, I., 2007. Data fusion of high-resolution satellite imagery and LiDAR data for automatic building extraction. ISPRS Journal of Photogrammetry & Remote Sensing 62 (2007) pp.43–63, 15 February 2007

Vosselman, G., 1999. Building reconstruction using planar faces in very high density height data. International Archives of Photogrammetry and Remote Sensing 32 (Part 2), 87–92.