ASPRS Annual Conference, Denver, CO, May 23-28, 2004

Building Segmentation and Regularization from Raw Lidar Data

Aparajithan Sampath   Jie Shan

Geomatics Engineering
School of Civil Engineering
Purdue University
550 Stadium Mall Drive
West Lafayette, IN 47907-2051

ABSTRACT

This paper presents an approach to process raw lidar data for detecting, segmenting and regularizing buildings. By regularizing, we mean the extraction of the exact and realistic footprint of a building. In the first step we separate the 3D point cloud into two major classes, namely the building point dataset and the non-building point set. This was accomplished by using a one dimensional filtering technique that makes use of the slope and height information between two consecutive lidar points in the point cloud. In the second step, individual buildings were then segmented from the building point dataset using a region growing algorithm, where we use a 3 dimensional window around a selected point and then move the window till we select all points that belong to the building. This process is carried out till all the building points are segmented. In the third step, we determine the building boundary points with an adapted convex hull formation approach. These points were then used to determine parametric equations for lines that represent the building edge. In the final step, a least squares model is used to square or regularize the building edges to form the final building footprints. Buildings over Baltimore downtown are used in our study. Their quality is evaluated by comparing the lidar generated results with ortho images.

INTRODUCTION

3D point clouds generated using Light Detection and Ranging (Lidar) can be used to model urban environments and generate city models. A dense point cloud can be used to accurately measure various features, both natural and man made. The task, then, is to extract these features from this dense dataset.

Lidar combined with GPS (Global Positioning System) and IMU (Inertial Measurement Unit) generates dense three-dimensional (3-D) georeferenced coordinates for the entire reflected terrain surface. We can easily generate 3-D topographic surface information using this data (Ackermann, 1999; Balsavias, 1999). The first step in generating city models is to remove all the points that do not represent buildings from the dataset. Since it is easier to mathematically define ground than other features, returns from ground are first separated from non-ground features. In this process, we can also generate bald ground DEMs of the area (Axelsson, 1999; Sampath and Shan, 2003; Schickler and Thorpe, 2001; Sithole, 2001; Vosselman, 2000; Vosselman and Mass, 2001).

In this paper we focus on segmenting and extracting buildings from raw lidar point data. The task of reconstructing buildings has been tackled in different ways. One approach is to subtract the bare ground DEM obtained after the separation of ground points from the Digital Surface Model (DSM). This will give the footprints of buildings. The problem of converting these footprints to vectors is addressed by assuming to orthogonal dominant directions for each building and then constraining the building edges to lie along those directions (Al-Harthy and Bethel 2002). Rottensteiner and Briese (2002) applies a morphological filter over the building footprints to get a binary image of planar regions. A connected component analysis then reduces these regions to smaller buildings. A region growing algorithm can be applied such that nearby pixels within a threshold are taken to belong to a particular building. Another approach to this problem would be to use the detected building points and the surrounding ground points to interpolate the building boundaries. This is achieved after determining the internal 3-D breaklines of the buildings (Morgan and Habib, 2002).

This paper presents an approach to process raw lidar data for detecting, segmenting and regularizing buildings. By regularizing, we mean the extraction of the exact and realistic footprint of a building. In the first step we separate

the 3D point cloud into two major classes, namely the building point dataset and the non-building point set. This was accomplished by using a one dimensional filtering technique that makes use of the slope and height information between two consecutive lidar points in the point cloud. In the second step, individual buildings were then segmented from the building point dataset using a region growing algorithm, where we use a 3 dimensional window around a selected point and then move the window till we select all points that belong to the building. This process is carried out till all the building points are segmented. In the third step, we determine the building boundary points with an adapted convex hull formation approach. These points were then used to determine parametric equations for lines that represent the building edge. In the final step, a least squares model is used to square or regularize the building edges to form the final building footprints. As a distinction, this approach does not need to fix any particular building direction. Instead, all line segments are subject certain levels of adjustment depending on their length. Buildings over Baltimore downtown are used in our study. Their quality is evaluated by comparing the lidar generated results with ortho images.

SEPARATION OF BUILDING FROM GROUND

An airborne lidar data set is used in this study. It covers the downtown Baltimore, Maryland. The average point density is one point per 5.5 square meters. It is noted that the point ground spacing distance within lidar profile (across track) is about 2.3 meters, while the ground spacing along fight direction is about 4.8 meters.

The first step to building segmentation is to separate ground points from building points in the given lidar point cloud. In the precious study [Sampath and Shan, 2003], we proposed a filtering approach to label points as ground and building points in two sequential steps along the 1-D lidar profiles. The labeling approach is based on slope and elevation assessment and is implemented along the lidar profile in two opposite directions. A local linear regression along the lidar profile is then followed to further remove the non-ground points remained from the labeling process. Tests over three urban areas with different complexity show that over 95% of the lidar points can be correctly labeled. For the details and the performance of the proposed approach we refer to [Sampath and Shan, 2003]. In the following studies, we will use the building point cloud separated from this step.

BUILDING SEGMENTATION

Once building points have been separated from the ground points, we further segment them such that each point is assigned to a unique building, i.e. segment the building point dataset into a set of points that represent a single building. For this objective, a 3-D region growing algorithm is applied, which is modified based on the approach reported in [Sampath and Shan, 2003]. This algorithm consists of the following steps:

1) Start from any building point P_0.
2) Center a 3-D window (cube) around the point and collect all the points $A = \{P_{1,} P_2, ...P_k\}$ that fall within the window.
3) Move the window to P_1
4) Collect the points that fall within the window and store them in a temporary set, say tempPoints=$\{tP_1, tP_{2,...}, tP_r\}$.
5) Move to point P_2 and place the window over it. Append the newly collected set of points to the variable tempPoints, and in the process making sure that no two points are the same.
6) Continue the process till the window has been placed over all the points $P_{1,} P_2, ...P_k$
7) Merge Points in A and points in tempPoints and store them in B. i.e. $B = \{B \cup A \cup tempPoints\}$. Initially B is a null set.
8) Replace Points in A with points in tempPoints such that the newly populated set A is equivalent to $\{A \cup tempPoints\}$.
9) Go back to step 3.
10) Stop when no new points are added to the set B.

In this way, building points are further segmented into points representing belonging to each individual building. The proposed segmentation approach has several distinct features. First of all, it does not require any special data structure. The initial input into the algorithm is a 3D point cloud, which is basically a set of X, Y and Z coordinates. Another advantage of this algorithm is that it does not need the points to be in any particular order, i.e. points close by spatially can be indexed anywhere in the input file. Also, the number of searches the algorithm has to make progressively reduces as more and more points are assigned building numbers. Another advantage of this algorithm is that it does not depend on the way in which building points have been separated from ground points in the raw lidar dataset. All it needs as an input is the building points, regardless of the way in which they are generated. Figure 1 shows a portion of the segmentation results over the test area.

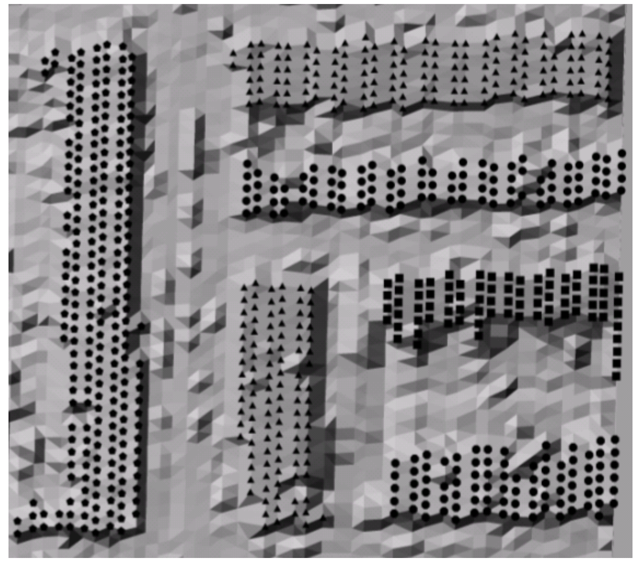

Figure 1. Segmented buildings shown with different symbols

BUILDING BOUNDARY TRACING

Once the points representing a single building has been segmented from the building point dataset, the next step is to determine the footprints of the buildings. An example of the point cloud representing a single building is shown in figure 2, which will be used an example to illustrate our tracing methods. The task was divided into two stages. In the first stage, the points which best represent the boundary of the building was separated from the above set of points. In the second stage, parametric lines were drawn to represent the boundary edges of the building.

To determine the boundary points, we used a modified form of the convex hull algorithm. When a set of coordinates are given as an input, the convex hull algorithm determines those coordinates that would constitute a convex hull. For instance, figure 2 shows a convex hull for the building point set shown in figure 2. Clearly, it does not represent all the boundary points of the building, nor does it bring out the shape of the building accurately. In this The algorithm determines the hull points by selecting the left-most point and then successively determining those points that make the least angle with the last generated convex edge. To accomplish this, the algorithm compares the slope of the last edge that is generated with lines formed by connecting the current point with the all other points.

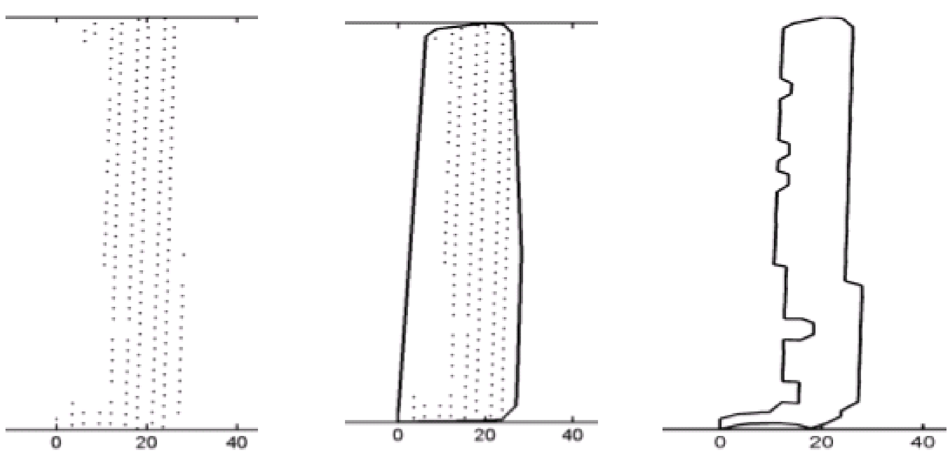

Figure2. Building points (left) and convex hull (middle) and boundary (right)

To determine the boundary points for our dataset, we adapted this algorithm. The steps are given below:

1) Start from the left most point P.
2) Select all points that lie within a threshold distance from this point, say $Pts = \{p_{1,} p_2, ... p_k\}$. The threshold distance was the point density of the dataset.
3) Determine the slopes of all the line segments that connect the current point P, with the set of selected points in Pts.
4) Determine the point $\mathrm{p_r}$ in Pts which has the least angle from the Y-axis.
5) Move to this point and consider this to be the next current point $P_c^{'}$ and determine the slope of the line (say L_c) connecting this point with the previous point.
6) Select all points that lie within the threshold distance from $P_c^{'}$, determine the point/line segment which makes the least angle with line L_c . Append this point to the boundary point set and make this point the new $P_c^{'}$.
7) Continue till the first selected point is reached.

The result is shown in figure 2. The central idea was to use and adapt a convex hull existing algorithm to determine the boundary points. The difficulty in directly applying the convex hull algorithm is that buildings are not necessarily in convex. But, if we restrict the search space of the algorithm to a smaller area, given by a threshold distance, the algorithm will give us an accurate outline of the border points on a building. As can be expected, the result of the algorithm depends on the threshold distance that is assigned in steps 2 and 6. This distance, in turn, is proportional to the point density or the point spacing of the 3D point cloud. In most point clouds, the distance between two points lying on (or parallel) the X-axis will be different from the distance between two adjacent points lying on(or parallel) to the Y axis. Therefore the threshold distance mentioned in steps 2 and 6 should be adjusted accordingly. The convex hull algorithm was used as we found that it is easy to implement as well as to adapt for our purposes.

BUILDING SQUARING

The result that is obtained from the above set of steps is a set of connected points that are at the building boundary. In the next stage, parametric lines in two mutually perpendicular directions were made to fit the boundary edges. For this, it is assumed that we are dealing with buildings that are not curved and their edges are parallel or intersect ninety degrees. It is clear from just a glance at figure 2 that not all the lines are exactly perpendicular to each other. But it can be noted that the longer line segments that can be extracted are always in mutually perpendicular to each other. These larger lines represent the basic frame of the building footprint and the directions of these lines can be taken to represent the direction of the buildings. This idea is the basis of our method to determine the footprint of a building and a global solution based on the least squares criterion is proposed to square the building boundary so that the extracted buildings are regularized.

There exists a one-to-many correspondence between boundary edges and the boundary points. Each point lies on a single line, unless it is a point that lies where two edges intersect. By following the boundary points sequentially, we collected different sets of points, each of which corresponds to an edge. This was done by looking for positions where the slope between two consecutive points e.g., $\left(P_i, P_{i+1}\right)$ is different from $\left(P_{i+1}, P_{i+2}\right)$.

The set of points $A = \{(p1_{1,} p1_2, ... p1_{k1}), (p2_{1,} p2_2, ... p2_{k2}), ...(pn_{1,} pn_2, ... pn_{kn})\}$

is mapped to the line segments $\{l_1, l_{2,} ....., l_n\}$

Then, the longest set of these lines (e.g., $\{l_i, l_j .. l_m\}$) were selected, along with their corresponding sets of points

$(say\ \{(pi_{1,} pi_2, ... pi_{k1}), (pj_{1,} pj_2, ... pj_{k2}), .., (pm_{1,} pm_2, ... pm_{km})\}$).

In the third step, the least squares solutions for these lines are determined, with the constraint that the slopes of these lines are either equal (the lines being parallel), or their product is equal to -1 (in which case, the lines are perpendicular). The solutions consist of a set of parameters that describe each of the line segments $\{l_i, l_j..l_m\}$. The equation of the lines that we had used is *Ax+ By +1=0*. In particular, we have the following set up for our building squaring problem. For each line segment

$$A_i x_{ij} + B_i y_{ij} + 1 = 0 \qquad i = 1,2...n; \quad j = j(i) = 1,2,....m_i \tag{1}$$

where n is the number of line segments, m_i is the number of points on line segment i. Line segments of a building are grouped based on their slope. Lines that are parallel within a given tolerance are sorted as one group with the same slope. Let K be the number of parallel line groups, then lines in every group should meet the following condition

$$\frac{A_s}{B_s} + M_k = 0 \qquad k = 1,2,...,K \quad s = s(k) = 1,2,....,n_k \tag{2}$$

where M_k is the slope of parallel line group k, n_k is the number of lines in the k-th parallel line group. Similarly, for the line groups that are perpendicular, we can write the following condition equation

$$M_u M_v + 1 = 0 \qquad u,v = 1,2,3....,K; \; u > v \tag{3}$$

The least squares criterion is used to solve the above equation systems. The unknowns include all the line segment parameters A_i and B_i $(i = 1,2,...,n)$, and the slope $M_k\,(k = 1,2,...,K)$ of parallel line groups. In the current study, only two groups of parallel lines, namely horizontal and vertical line segments are considered. This leads to only one conditional constraint in Equation (3).

To determine the parametric line segments, a hierarchical approach is designed. This approach starts with relatively longer line segments detected in the lidar points. In the next step, relatively shorter line segments are introduced and their parameters are determined based on the slopes of the line segments obtained from the previous step, keeping in mind that we consider only two possible directions for each line segment.

Figure 3 shows the determined parametric line segments for the building boundary. The final squared building and the original building points are also shown in figure 3, with the original lidar points overlaid atop.

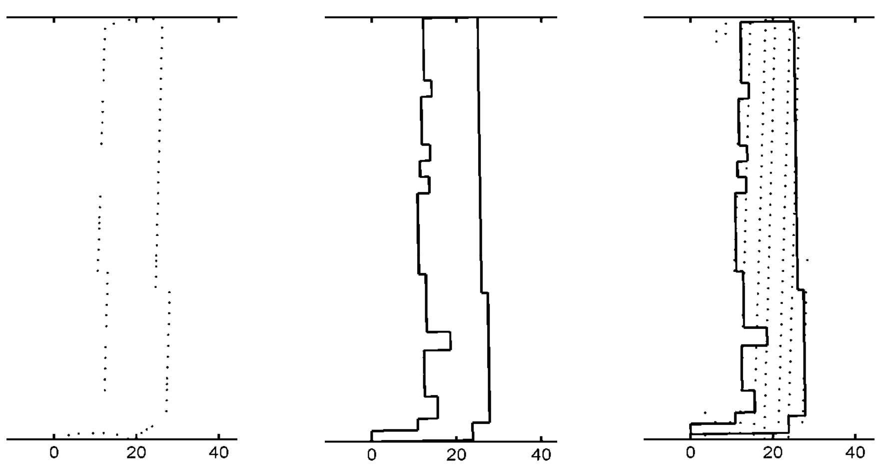

Figure 3. Boundary points (left), parametric lines (middle) and the squared result overlaid with original lidar points (right) .

There are several distinctions of this least squares based hierarchical building squaring approach. First, it is robust to possible errors in building segmentation, boundary tracing. This is because of the hierarchical implementation of the solution, shorter line segments are processed after longer lines. Second, the errors of final extracted building can be evaluated through the least squares adjustment process, using either the residual values between estimated coordinates and the observed coordinates of the points or determining the distance of each of these points from the parametric lines. Third, it provides a global optimization solution to the building squaring problem. No points or line segments are taken as fixed reference. The longer line segments receive larger weights than shorter ones. All points and line segments are subject to certain adjustment in position depending on their contribution to the line segments.

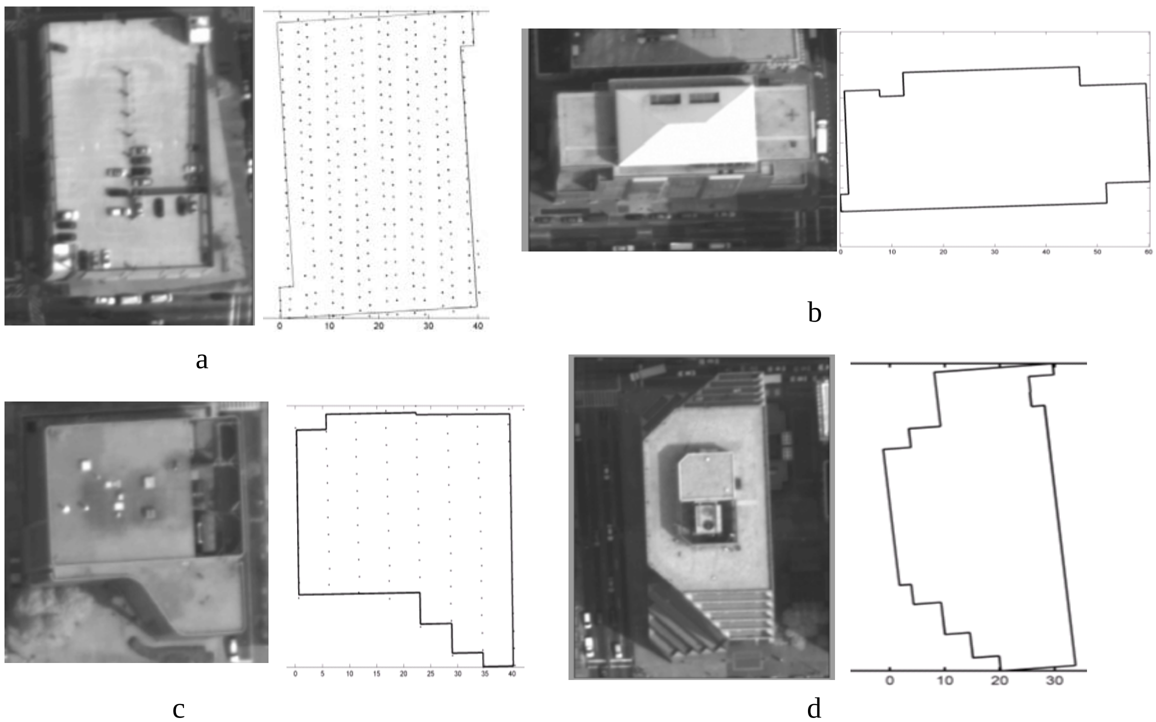

Figure 4. Examples of regularized buildings with their ortho images

Figure 4 present the results of several squared buildings along with their orthoimages. They are obtained from the above described regularization process. It can be seen that squared results are a quite accurate representation of the buildings. Resultant edges reach an optimal fit with the detected building points and delineated building boundaries. At the building boundary, the parametric edges reach the best balance of the zigzagged lidar point sequence. This conclusion indicates the efficiency of the proposed approach for the rectilinear buildings. It also supports our hierarchical solution strategy, where all line segments are subject to adjustment with longer ones being adjusted little and shorter ones larger. This strategy ensures our solution to be robust to the lidar data resolution and the possible non-building points mistakenly included in precious processing steps.

Step-effect will appear if this regularization approach is inappropriately used to square non-rectilinear buildings. As is shown in Figure 4b and 4c, edges that are neither parallel nor perpendicular to the rectilinear boundaries will be formed to a step shape. This can be interpreted as a rectilinear approximation to the realistic building edges of any direction. The higher the lidar data resolution, the better the non-rectilinear edges can be approximated. This can be illustrated by the upper left and lower right slant edges in Figure 4d and curved edge in lower middle of Figure 4c.

The squared building shape is affected by the segmentation results. In our approach, individual buildings are segmented by a 3D growing approach. This indicates a building is not only defined by its 2D (footprint) connectivity, but also its 3D (vertical) connectivity. For buildings with large vertical differences, their 3D segmentation results may be different from the ones from 2D segmentation. An example of such case is shown in Figure 4d, where its lower left part actually is not segmented as part of the building due to its large vertical difference with the main building body. As a result of this vertically biased segmentation, the proposed building squaring process may "step" the non-rectilinear edge as shown in Figure 4d. Although higher resolution of lidar data may amend this step effect, further study is needed that considers both 2D and 3D discontinuities in the building segmentation process

CONCLUSIONS

In this research, our objective is to develop a series of steps to segment buildings from lidar point cloud. We have used only the lidar datasets, and not any other auxiliary information, such as building foot plans. It is also our intention to devise a methodology which avoids interpolating the existing datasets and using complex data structure. A novel labeling approach to initially classify the given point cloud into ground points and building points is introduced and proven effective.

The building detection and segmentation algorithm makes use of the fact that two spatially different groups of point clouds representing two separate buildings will be at some distance greater than the point spacing of the original lidar dataset. We proposed a region growing algorithm based on a 3D cube which searches for nearby points lying on the same building.

A least squares based hierarchical building squaring approach is introduced. This algorithm suggests steps which determine the footprint of a building as a series of line segments that are parallel and perpendicular to each other. Since shorter line segments are processed after longer lines, errors from previous steps are minimized. In this approach, no line segment is chosen as fixed and all are subject to certain levels of adjustment in direction and position, depending in general on the length of the line segment. Such hierarchical strategy ensures our solution to be robust to the lidar data resolution and the possible non-building points mistakenly included in the precious steps.

Our experience shows that a reliable segmentation is necessary for a quality building squaring outcome. Buildings with more than two principal directions and non-rectilinear edges need certain modification and adaptation of the reported hierarchical strategy.

REFERENCES

[1]. Ackermann,F., 1999. Airborne Laser Scanning – Present Status and Future Expectations, ISPRS Journal of Photogrammetry & Remote sensing, Vol.54, Issues 2-3, July, pp.64-67.

[2]. Al-Harthy, A., Bethel, J, 2002. Heuristic Filtering and 3D Feature Extraction, ISPRS Commission III, September 9-13, 2002, Graz, Austria

[3]. Axelsson, P., 1999. Processing of laser scanner data- algorithms and applications, ISPRS Journal of Photogrammetry & Remote sensing, Vol.54, Issues 2-3, July, pp138-147.

[4]. Baltsavias, E., 1999. A Comparison Between Photogrammetry and Laser Scanning, ISPRS Journal of Photogrammetry & Remote sensing, Vol.54, Issues 2-3, July, pp.83-94.

[5]. Morgan, M., Habib, A., 2002. Interpolation Of Lidar Data And Automatic Building Extraction, *ACSM-ASPRS 2002 Annual Conference proceedings*

[6]. Rottensteiner,F., Briese, C.,2002, A New Method For Building Extraction In Urban Areas From High-Resolution Lidar Data, *ISPRS Commission III*, September 9-13, 2002, Graz, Austria

[7]. Sampath, A., and J. Shan, 2003. Building Segmentation from Raw Lidar Data, *ASPRS Annual Conference, Anchorage*, AK, May 3-9.

[8]. Schickler, W., Thorpe, A., 2001. Surface estimation based on Lidar, CD-ROM Proceedings of *ASPRS Annual Conference*, April 23-27, St. Louis, MO.

[9]. Sithole, G. 2001. Filtering of Laser Altimetry Data Using a Slope Adaptive Filter In: Hofton, Michelle A. (Editing), Proceedings of the ISPRS workshop on Land Surface Mapping and Characterization Using Laser Altimetry, Annapolis, Maryland, *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Science*s, Vol. XXXIV part 3/W4 Commission III, pp.203-210.

[10]. Vosselman, G., 2000. Slope Based Filtering of Laser Altimetry Data, International Archives of Photogrammetry and Remote Sensing, Vol. XXXIII PartB3, Amsterdam 2000, pp.935-942.

[11]. Vosselman, G., Mass, H., 2001. Adjustment and Filtering of Raw Laser Altimetry Data, *Proceedings OEEPE Workshop on Airborne Laserscanning and Interferometric SAR for Detailed Digital Elevation Models,* March 1-3, Stockholm.