

AUTOMATIC SEGMENTATION OF LIDAR DATA AT DIFFERENT HEIGHT LEVELS FOR 3D BUILDING EXTRACTION

S M Abdullah, Mohammad Awrangjeb and Guojun Lu

Gippsland School of Information Technology, VIC 3842, Australia
{sm.abdullah, mohammad.awrangjeb, guojun.lu}@monash.edu

ABSTRACT

This paper presents a new LiDAR segmentation technique for automatic extraction of building roofs. First, it uses a height threshold, based on the digital elevation model to divide the LiDAR point cloud into ‘ground’ and ‘non-ground’ points. Then starting from the maximum LiDAR height, and decreasing the height at each iteration, it looks for coplanar points to form planar roof segments. At each height level, it clusters the points based on the distance and finds straight lines using the points. The nearest coplanar point to the midpoint of each line is used as a seed and the plane is grown in a region growing fashion. Finally, a rule based procedure is followed to remove planar segments in trees. The experimental results show that the proposed technique offers a high building detection and roof plane extraction rates while compared to a recently proposed technique.

Index Terms— Building extraction, detection, height levels, LiDAR.

1. INTRODUCTION

Building detection and 3D roof reconstruction has been an active research topic for more than two decades. Accurate detection of building boundary and roof planes can be a vital source of information for various applications, including urban planning, virtual reality, disaster management, detection of unlawful extension of properties. The possible areas of applications are increasing day by day. Hence, many automated algorithms have been reported over the last couple of decades. Those algorithms can be categorized into three main groups [1]: image only, LiDAR only and combination of image and LiDAR information.

Many researchers have applied 2D or 3D information from the photogrammetric imagery for detection and extraction of buildings. There are many strong reasons not to use image only method. In [2] the authors showed that, a high resolution image contains more detail information, so the complexity of detecting buildings from the non-buildings objects increases as the resolution of the image increases. Regardless of this problem, shadows and occlusions have also negative effect for building detection. The 3D information derivation

from stereo images, like depth information is even more challenging. Moreover, nearby trees of similar height make the use of such range data difficult [3].

In LiDAR only, LiDAR (Light Detection And Ranging) point cloud is used for building detection and extraction. The current LiDAR point cloud is dense, it has high accuracy in height and can be used to directly extract the three dimensional objects on the earth. Unlike image based method, LiDAR data is active system, so it can be gathered during the day or night. LiDAR data has other advantages like, fast data acquisition, high point density, canopy penetration and the minimum of ground truth [4].

Currently many researchers are trying to combine high resolution imagery and LiDAR data for the purpose of detection and extraction of buildings [1]. By using dense LiDAR data it is possible to avoid problems involving imagery alone.

In this paper we have focused on the building detection and extraction process based on LiDAR data only. LiDAR data is chosen for its highly accurate three dimensional information and less complexity in information processing. The rest of the paper is designed as follows: Section 2 discusses related research works for building detection and extraction using LiDAR data. Section 3 describes the proposed method, followed by the experimental results, analysis and comparison in section 4. Concluding remarks are then provided in section 5.

2. RELATED WORKS

Awangjeb and Fraser [5] presented a rule-based approach using LiDAR point cloud. They classified the raw LiDAR point cloud into ground points and non-ground points. A building mask was generated using the ground points and individual buildings and trees were obtained as clusters of black pixels from the mask. The co-planarity of each non-ground point was tested using Delaunay neighborhood. After that, planar segments were extracted from the non-ground LiDAR points. To refine the results rule-based approach was introduced. Finally, false planes were removed to get the final set of roof planes. Sampath and Shan [6] presented a solution framework for building roof extraction. They determined the planarity of each LiDAR point based on eigenvalue analysis. Non-planar

points were not consider for further processing. After that, they clustered the planar points by using fuzzy k-means approach based on their surface normals. Good evaluation results for both segmentation and reconstruction were achieved. However, the method exhibited high reconstruction error due to removal of LiDAR points near the plane boundaries. Moreover, the fuzzy k-means clustering algorithm is computationally expensive. Sohn et al. [7] proposed a Binary Space Partitioning (BSP) tree model for polyhedral building reconstruction. Buildings were detected by classifying the LiDAR data into building and non-building points. These classified buildings were clustered based on height and planar similarity. Then they extracted lines and generated building model by using BSP tree. Perera et al. [8] proposed an automated method for 3D roof outline generation. They introduced cycle graph for the best use of topological information. There method segmented LiDAR data into planar patches and by using it roof planes were extracted. Lines were extracted and subsequently used to construct a roof topology graph. Inner and outer boundaries were reconstructed by using this topology graph. Experimental results showed that, the process failed to detect some flat and shaded buildings while it included vegetation as part of the extracted buildings.

3. PROPOSED APPROACH

The proposed method starts with LiDAR data as input. It uses raw LiDAR data without any modification. LiDAR is a series of points stored as X, Y and Z. The X and Y values are easting and northing and Z is the elevation in metre.

Figure 1 outlines the proposed work. The algorithm has six major steps. Firstly, LiDAR point cloud is divided into two groups: namely ‘ground’ points and ‘non-ground’ points. As the method focuses on buildings, only the ‘non-ground’ points from the LiDAR point cloud are sufficient. Secondly, each of the LiDAR point is marked as coplanar or non-coplanar based on coplanarity analysis of each point and its neighborhood. Thirdly, lines are extracted from the LiDAR point cloud at different height levels. Considering the nearest coplanar point to the middle of a line as a seed point, a planar segment is grown in a region growing fashion. In the next step planes on trees and other non-building structures are removed using a rule-based procedure. Finally, the planar segments are combined on a neighborhood basis to obtain individual building boundary.

3.1. Finding non-ground points

The test area may have variation in elevation, so the ground height from the digital elevation model (DEM) is used for each LiDAR point. A DEM has been generated from the input LiDAR data by using commercial software like: MARS 7 [9]. To group the LiDAR point cloud as ‘ground’ points and ‘non-ground’ points, the ground height value from the DEM

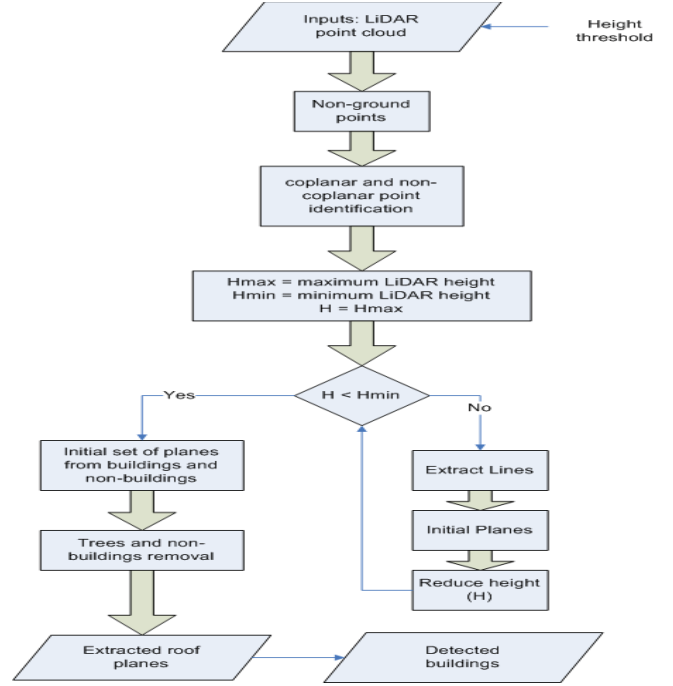
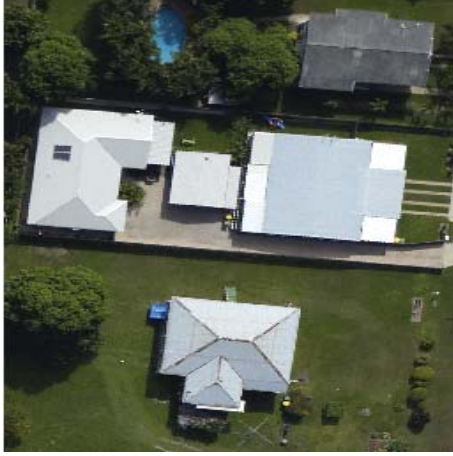


Fig. 1. Flow diagram of the proposed approach.

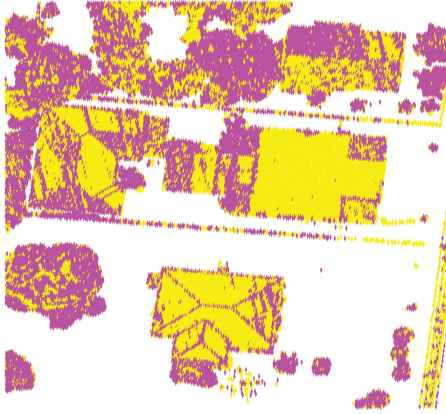
along with a threshold is used. The threshold value is considered here 1m [5]. So for each LiDAR point $T_h = H_g + 1m$ is calculated where H_g is the ground height. We consider any thing which is 1m higher than the ground height as building. So the algorithm is capable to detect small height buildings. If the ground height (H_g) is not defined in the DEM, the mean height from the surrounding neighborhood is used as the ground height. All the points that are above T_h are ‘non-ground’ points and the rest are ‘ground’ points. From now on, LiDAR data refers in this paper contains only the ‘non-ground’ points.

3.2. Coplanar and non-coplanar points

The eigenvalues of the covariance matrix is used to evaluate a point as coplanar or non-coplanar with respect to its neighboring points. At first, the LiDAR points are used to create a Delaunay triangulation. For each LiDAR point, the neighborhood triangles of it are determined. Each LiDAR point and its neighborhood triangles are considered to coplanarity analysis. If l is a LiDAR point and $dnbr(l)$ defines the neighborhood triangles of l (the triangles which contain l as one of its vertex) then these points are used for coplanarity analysis. Theoretically the eigenvalues from these points must be zero if l is a planar point regarding its neighborhood. But due to inherent noise associated with the LiDAR data, it is unlikely that, the eigenvalues will be zero, even for planar neighborhood. So normalized eigenvalue is used instead of using individual eigenvalue. If the normalized eigenvalue is no more than



(a) A test area



(b) Planar and non-planar points

Fig. 2. Coplanar and non-coplanar points for figure (a).

0.005 [6] the LiDAR point is considered as coplanar with respect to its neighborhood. Figure 2(b) shows the coplanar and non-coplanar points for a data set. The points which are marked by yellow colour are coplanar points and the remaining magenta coloured points are non-coplanar. As can be seen the majority of points on trees are non-coplanar and those on buildings are coplanar.

3.3. Line extraction

This step starts with maximum LiDAR point height. Accumulate all of the points at this particular height with a tolerance height threshold. This can be represented as $M = \{p \in P | H - \tau \leq p_h \leq H + \tau\}$, here P represents the LiDAR point cloud where p_h represents the height of point p and H represents the current height level. Points satisfying the above equation are selected where τ indicates the tolerance height threshold, which is set here 10 cm. As LiDAR data has inherent noise, it is unlikely to get all the points on a

particular height level. τ enables to accumulate more points on a height level where the elevation error of LiDAR data is more than 10cm. All of the points in M may not belong to the same building or plane. The points in M are clustered based on two dimensional (X,Y) distance. Each of the cluster is used to extract lines. The extracted lines are used to select the seed point for a plane as discussed below. This step runs iteratively and in each iteration H is reduced by 0.5m until it reaches the minimum LiDAR point height. This value (0.5m) is set empirically. If the density of the LiDAR point cloud is low ($\leq 1 \text{ point/m}^2$), then the chance of getting new seed point for plane extraction will be eventually low. However, for higher density the possibility increases. Setting the value as 0.5m allows us not to lose any significant plane.

3.4. Initial plane extraction

The findings of all of the previous steps are used here to extract planar segments. We describe the procedure for a single plane extraction. All of the planar segments can be found by running the procedure iteratively. A coplanar point which is the nearest to the middle of a straight line is selected as the seed point. The seed point and its neighboring points are used to generate the plane equation. Now the initial plane is grown until new points can be added. To grow the initial plane those points are chosen which are the neighborhood of the current plane and not belong to it. Before updating the plane equation, the new points are filtered based on plane fitting error and height difference between the estimated height and the LiDAR height of the points. Two different threshold are chosen for this purpose. The values are set 15 cm and 10 cm respectively [10]. All the new points which satisfy one of the two threshold values are added to the initial plane and the plane equation is updated. In this region growing fashion all of the planes are extracted. By using raw LiDAR data the possibility of growing non-building planes are higher. To reduce the number of unwanted planes the average standard deviation of height of all of the LiDAR points is used as a reference to determine a plane to be useful or unwanted. Comparatively, planes on trees have higher standard deviation than planes on building. If the standard deviation of height of a plane is higher than the average standard deviation the plane is discarded. By using this criteria some of the unwanted planes (planes on trees) can be discarded before final pruning. Figure 3 illustrates all of the initial planes after this step for a test data set.

3.5. Tree and non-building plane removal

All of the planes extracted in the previous step are not required for building detection. Some of the planes are extracted on trees and other non-building structures. To remove these unwanted planes a rule-based approach is applied to the initial set of planar segments.



Fig. 3. Extracted planar segments after initial plane extraction step.

3.5.1. Used point ratio

The ratio of used points and actual number of points in a plane is considered as a parameter for tree removal. To calculate the actual number of points, the planar segment is bounded by a rectangular region. All the LiDAR points inside it is considered as actual points. If the ratio is less than 60% the plane is removed.

3.5.2. Area and width calculation

For each plane the surrounding area and width is calculated. If a plane has negligible width ($\leq 1m$) regardless of other parameters the plane is removed. In plane extraction step some planes are extracted on fence around the building, which are not considering as part of the building. The length of the plane may be large but it has small width. This test removes those planes. The area of a plane is also considered to remove unwanted planes. For an isolated plane (single plane building) if it has small area ($\leq 3m^2$) the plane is not considered as a building plane and for non-isolated planes the area parameter is reduced to $1m^2$.

3.5.3. Height gap

The above two rules are able to remove most of the planes on trees and non-building structures. To remove the remaining ones, the height gap within a plane is evaluated. In this measurement, the main concern is for finding any significant height difference in a plane. All of the points in a plane are clustered based on height. If there are several clusters in a

plane, the average height difference for all clusters are calculated. If there has a significant difference in height, the plane is considered as tree plane. The motivation of this approach is, for building plane, there are less height gap as the LiDAR points are not penetrating the roof. But for the trees, as the LiDAR points are coming from various parts of the tree there may have vertical height gap in the plane. This observation is used here to remove planes on trees. Figure 4 shows a portion of data set where the left image presents the initial situation before applying tree removal process and the right one shows the effect after tree removal.

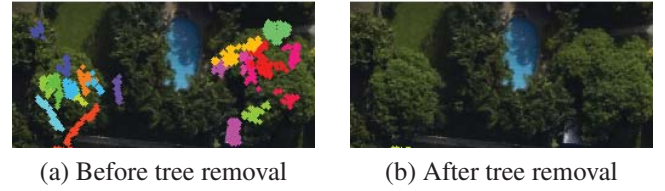


Fig. 4. A portion of a data set showing the situation before and after tree removal step.

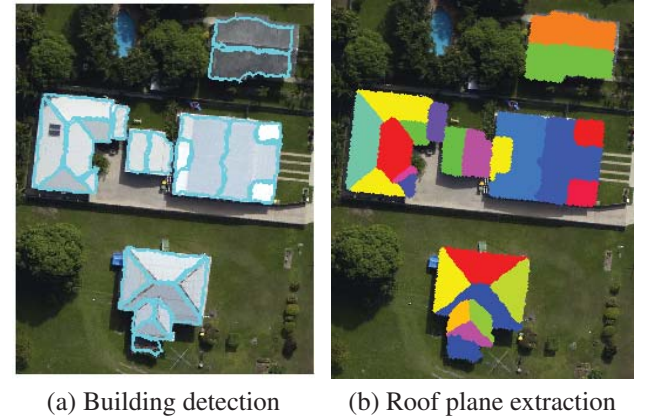


Fig. 5. Final detected buildings and extracted roof planes.

3.6. Building outline detection and plane extraction

This is the final step of the algorithm, where the outline of the building is generated. The remaining planes are grouped into several clusters based on distance. Each of the cluster represents a building. By combining all the points in a cluster the building outline is obtained. After that the boundary of each building and each plane is calculated. The final detected buildings and extracted roof planes are shown in figure 5 for a test data set. Images are used only for visualization purpose. The image data is not used in the algorithm.

Table 1. Evaluation results for building detection: Completeness (C_m), Correctness (C_r) and Quality (Q) for per-area, per-object and object over $50m^2$ in percentage.

Scene	C_{ma}	C_{ra}	Q_a	C_{mo}	C_{ro}	Q_o	$C_{mo,50}$	$C_{ro,50}$	$Q_{o,50}$
Area 1	92.3%	88.4%	82.3%	86.5%	94.1%	82.1%	100.00%	100.00%	100.00%
Area 2	93.2%	90.1%	84.6%	85.7%	85.7%	75.0%	100.00%	91.7%	91.7%
Area 3	94.4%	85.0%	80.9%	80.4%	91.5%	74.8%	97.4%	97.5%	95.1%
Average	93.3%	87.9%	82.6%	84.2%	86.8%	77.3%	99.1%	96.4%	95.6%

Table 2. Evaluation results for roof plane extraction: Completeness (C_m), Correctness (C_r) and Quality (Q) for per-area, per-roof plane and roof plane over $10m^2$ in percentage.

Scene	C_{ma}	C_{ra}	Q_a	C_{mr}	C_{rr}	Q_r	$C_{mr,10}$	$C_{rr,10}$	$Q_{r,10}$
Area 1	88.8%	93.0%	83.2%	74.3%	93.1%	70.4%	86.4%	94.3%	82.1%
Area 2	87.9%	93.8%	83.1%	72.5%	96.2%	70.4%	91.7%	97.9%	89.9%
Area 3	91.6%	89.6%	82.8%	83.0%	94.3%	79.0%	94.4%	95.0%	89.9%
Average	89.4%	92.1%	83.0%	76.6%	94.5%	73.2%	90.8%	95.7%	87.3%

4. EXPERIMENTAL RESULT ANALYSIS

Experiments run on ISPRS (International Society for Photogrammetry and Remote Sensing) benchmark data set, provided by ISPRS WG III/4. The data set was captured over Vaihingen in Germany which has three different test areas. Area 1 is situated in the center of the city of Vaihingen. It is characterized by dense development consisting of historic buildings having rather complex shapes, but also has some trees. The size of the area is $217m \times 161m$ with point density $3.5/m^2$. Area 2 is characterized by a few high-rising residential buildings that are surrounded by trees and has an area of $231m \times 203m$ with point density of $3.9/m^2$. Area 3 is a purely residential area with small detached houses. It has area of $235m \times 168m$ with point density of $3.7/m^2$. Figure 6, 7 and 8 shows the building detection and roof planes extraction results. Correctly classified buildings and planes are marked with yellow colour (true positive), misclassified buildings and planes are marked with red (false positive) and missing buildings and planes are marked by blue (false negative).

Table 1 shows the experimental results of building detection on these test areas. It shows completeness, correctness and quality for per-area and per-object level. The first three columns represent the completeness, correctness and quality for pixel based evaluation. The pixel size is consider $0.5m$. The outcome shows a consistent result with an average of 93% correctness. The next three columns show the result of object-based evaluation and the remaining shows evaluation result for object-based for more than $50m^2$ area.

Roof plane extraction results are shown in Table 2. The pixel-based (area-based) experiment is carried out with $0.1m$ pixel size. The result shows that it misses some of the building points, some parts of buildings may have height lower

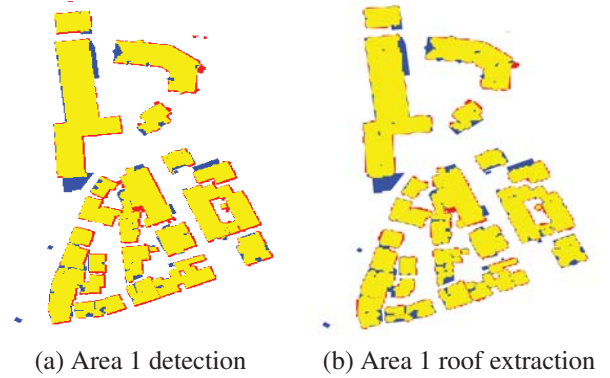


Fig. 6. Buildings detection and roof extraction of area 1.

than $1m$ from the ground height, so the average completeness is about 89%. But it includes less trees, hence the average correctness is above 92%. As it misses some LiDAR points on roof planes so per-roof completeness is reduced and the average completeness is about 76%, but as the same nature of pixel-based it includes few trees points, so the average correctness is higher than 94%. For large planes ($10m^2$) it exhibits sound result with around 91.0% completeness and more than 95% correctness.

4.1. Comparison result

The results are compared with a recent work of Awrangjeb et al. [5] For building detection, the object-based completeness is higher or equal for all test areas with greater average completeness. Two from the three test areas have higher completeness in pixel-based. For roof plane extraction, the object-

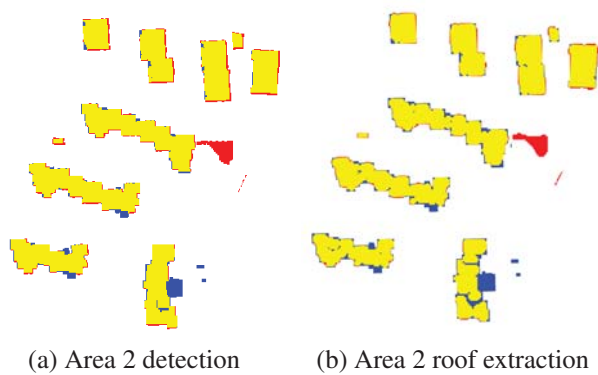


Fig. 7. Buildings detection and roof extraction of area 2.

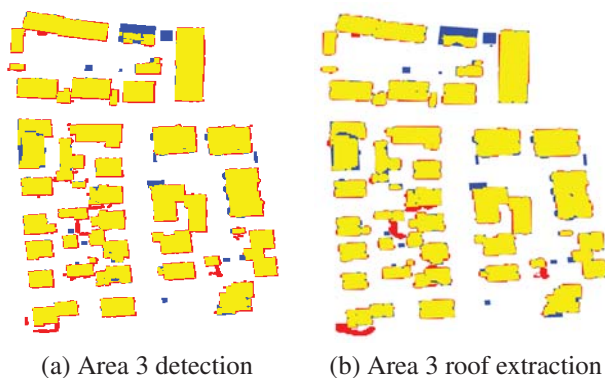


Fig. 8. Buildings detection and roof extraction of area 3.

based average completeness is approximately equal however the average correctness increases from 89.7% to 94.5%. Consistent results are found for large roof planes ($> 10m^2$) where the average correctness boost up by 4.3%.

5. CONCLUSION

In this paper, a complete method for building detection and roof plane extraction using LiDAR point cloud is presented. By using LiDAR data only, the complexity of the algorithm is reduced and we get rid of using image. The experimental results show that the algorithm works fine with different test cases. The algorithm is also tested with non benchmark data set and it shows consistent result. In most of the situation the comparison results are in favor of the proposed method. However, we need to make the algorithm more robust so it can identify small height buildings and include less vegetation.

6. REFERENCES

[1] Mohammad Awrangjeb, Chunsun Zhang, and Clive S. Fraser, "Automatic reconstruction of building roofs us-

ing lidar and multispectral imagery," in *DICTA*, 2011, pp. 370–375.

[2] L. Cheng, J. Y. Gong, X. L. Chen, and P. Han, "Building boundary extraction from high resolution imagery and LIDAR data," in *ISPRS Congress*, 2008, pp. 693–698.

[3] D. H. Lee, K. M. Lee, and S. U. Lee, "Fusion of lidar and imagery for reliable building extraction," *Photogrammetric Engineering and Remote Sensing*, vol. 74, no. 2, pp. 215–226, Feb. 2008.

[4] Satari M., Samadzadegan F., Azizi A., and Maas H.-G., "A multi-resolution hybrid approach for building model reconstruction from lidar data," *The Photogrammetric Record*, vol. 27, pp. 330–359, Sept. 2012.

[5] M. Awrangjeb and C. S. Fraser, "Rule-based segmentation of lidar point cloud for automatic extraction of building roof planes," *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. II-3/W3, pp. 1–6, 2013.

[6] Aparajithan Sampath and Jie Shan, "Segmentation and reconstruction of polyhedral building roofs from aerial lidar point clouds," *IEEE Trans. Geoscience and Remote Sensing*, vol. 48, no. 3-2, pp. 1554–1567, 2010.

[7] G. Sohn, X. F. Huang, and V. Tao, "Using a binary space partitioning tree for reconstructing polyhedral building models from airborne lidar data," *Photogrammetric Engineering and Remote Sensing*, vol. 74, no. 11, pp. 1425–1440, Nov. 2008.

[8] S. N. Perera, H. A. Nalani, and H.-G. Maas, "An automated method for 3d roof outline generation and regularization in airborne laser scanner data," *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. I-3, pp. 281–286, 2012.

[9] MARS version 7, "The merrick advanced remote sensing (MARS) software," <http://www.merrick.com/Geospatial/Software-Products/MARS-Software>.

[10] Mohammad Awrangjeb, Chunsun Zhang, and Clive S. Fraser, "Automatic extraction of building roofs using lidar data and multispectral imagery," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 83, pp. 1 – 18, 2013.