Aerial 3D Building Detection and Modeling From Airborne LiDAR Point Clouds

Shaohui Sun, *Student Member, IEEE*, and Carl Salvaggio, *Member, IEEE*

***Abstract*—A fast, completely automated method to create 3D watertight building models from airborne LiDAR point clouds is presented. The proposed method analyzes the scene content and produces multi-layer rooftops with complex boundaries and vertical walls that connect rooftops to the ground. A graph cuts based method is used to segment vegetative areas from the rest of scene content. The ground terrain and building rooftop patches are then extracted utilizing our technique, the hierarchical Euclidean clustering. Our method adopts a "divide-and-conquer" strategy. Once potential points on rooftops are segmented from terrain and vegetative areas, the whole scene is divided into individual pendent processing units which represent potential building footprints. For each individual building region, significant features on the rooftop are further detected using a specifically designed region growing algorithm with smoothness constraint. Boundaries for all of these features are refined in order to produce strict description. After this refinement, mesh models could be generated using an existing robust dual contouring method.**

***Index Terms*—3D, building, LiDAR, vegetation, graph cuts, region growing, modeling.**

I. Introduction

THREE dimensional building reconstruction has been a highly active research topic for years. There has been an increasing demand in various applications such as urban planning, virtual tourism, computer gaming, real-time emergency response, and robot navigation. Commercial applications like Google Earth and Apple Maps have already deployed 3D building reconstruction techniques as a vital visualization component which has gained huge public acceptance. Models are usually created by texture mapping both aerial and ground-based images onto 3D geometric models. Traditionally, geometrical models are built up manually. There are some very capable, free tools like Google Sketch-Up, that allow the layperson to accomplish this, however, it does require a lot of human effort to create just a single building. It remains a very difficult and arduous task, especially when a large cityscape needs to be created. In the remote sensing community, there are several data sources that are suitable as input to the building reconstruction process. Imagery is the traditional and most available data source. Research on how to extract 3D information from ground or aerial imagery has been conducted for decades. Recent advances have enabled techniques to directly capture 3D information over large scale areas. With the emergence of LiDAR (light detection and ranging) technology, a powerful 3D representation in the form of a point cloud can been created to assist in the generation of 3D scenes in a more efficient and cost effective manner. Many modern techniques are developed relying on the input from LiDAR.

When collecting nadir-looking aerial data in an urban setting, the two scene components that dominate this data are the tops of man-made structures (rooftops) and the background terrain. The detection and segmentation of these rooftops from the terrain is hence a crucial step in building extraction. Often times, the outlines of these building rooftops are not simple geometric primitives, on the contrary, they can be quite complex making the model extraction process a challenging task. In this research, the key contribution is the fully automatic work flow that is able to exploit useful information to urban modeling from airborne LiDAR data only, effectively and robustly conducting the task of scene classification without spectral knowledge from optical image data and three dimensional building extraction and its geometric modeling. The process removes vegetations through a graph cuts based approach and detects the terrain and building footprints by an Euclidean clustering technique and then produces simplified, 3D meshing building models using solely LiDAR point cloud data for a large urban scene. One major challenging issue at this stage is how to efficiently and accurately segment building regions from the rest of the background regions in the scene, particularly vegetation, without the assistance of multi-spectral optical imagery. The graph cuts based approach designed in this research does a fine job on the scene classification. In addition, an effective method for the extraction of rooftop features is presented, by utilizing this method, a great amount of details on the rooftop are well maintained.

II. Related Work

There has been a plethora of work conducted for urban modeling from airborne LiDAR data, aerial images, or the combination. Hu *et al.* [1] described a few projects on the topic of large scale urban modeling. Musialski *et al.* [2] recently provided a more comprehensive overview of urban reconstruction from different perspectives which are not restricted to aerial data inputs. The complexity and difficulty of this problem has been approached in many ways, but the synergistic use of muli-modal datasets has become a prominent pedagogy in this research area. Sirmacek *et al.* [3] introduced a novel and robust approach for

Manuscript received September 28, 2012; revised February 12, 2013; accepted March 02, 2013. Date of publication May 14, 2013; date of current version June 17, 2013. This work was supported by the U.S. Department of Energy under BAA PDP08 Grant Number DE-AR52-07NA28115.

S. Sun and C. Salvaggio are with the Digital Imaging and Remote Sensing Laboratory, Chester F. Carlson Center for Imaging Science, Rochester Institute of Technology, Rochester, NY 14623 USA (e-mail: sxs4643@rit.edu; salvaggio@cis.rit.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/JSTARS.2013.2251457

1939-1404/$31.00 © 2013 IEEE

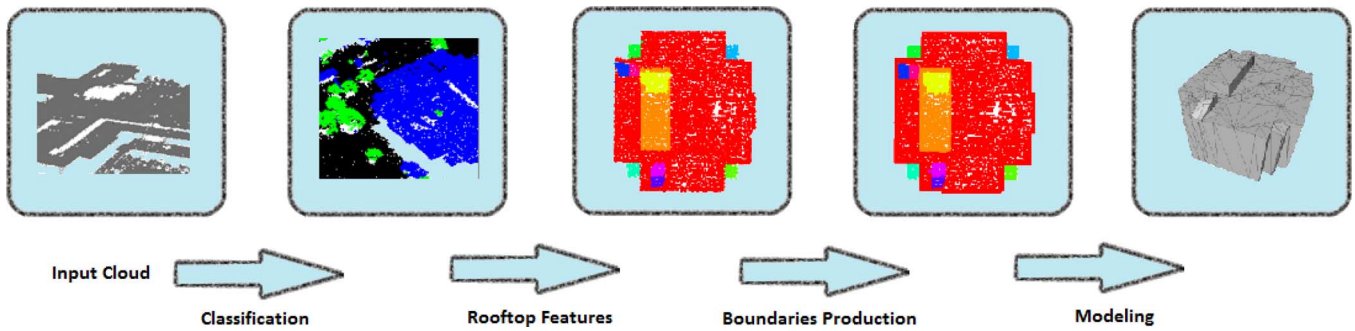


Fig. 1. Building detection and modeling work flow: classification identifies the terrain, trees and building footprints; significant planar features are detected on each individual rooftop; boundaries of all parts are produced; finally, the models are reconstructed by a volumetric method.

city modeling and also provided a detailed assessment of the algorithm performance on different types of remote sensed data, including both airborne and space-borne data.

Using 2D image data only, Moons *et al.* [4] proposed a method to reconstruct 3D polyhedral models. This method requires that accurate camera model information is readily available. Kim *et al.* [5] presented an approach to automatically describe a complex rooftop from multiple images. Image derived elevation data is used to assist feature matching. Once all of the 3D line features are obtained, the rooftop hypothesis generation process introduces the next important issue, time complexity. Nevatia *et al.* [6] introduced an automatic and interactive modeling method from aerial images, and Rau *et al.* [7] presented a Split-Merge-Shape algorithm for 3D building modeling in which an accurate scheme for 3D roof-edge measurements is proposed, however, these two ideas are semi-automatic, still requiring manual intervention.

Haala *et al.* [8] point out "the difficulties of aerial image interpretation also motivated the increasing use of three-dimensional point clouds from laser altimetry as an alternative data source".

Using LiDAR data only, Wang *et al.* [9] developed a Bayesian method to detect building footprints automatically from the LiDAR data. The point cloud has to be first segmented in to buildings, trees and grass as a pre-processing step. Verma *et al.* [10] presented a method to detect and construct complex buildings using LiDAR. It makes no assumption about prior knowledge about the types of buildings, but it doesn't address how to handle the situation if the rooftop has multiple layers/planar surfaces. Dorninger *et al.* [11] proposed a comprehensive approach for automated determination of 3D city models from LiDAR point cloud data, but their approach involved an interactive initialization called the coarse selection of building regions. Poullis *et al.* [12] addressed a rapid modeling approach for large-scale area reconstruction by using statistical considerations for segmenting the buildings. Zhou *et al.* [13] proposed a workflow for automatic building reconstruction from airborne LiDAR data. The highlights of this method are SVM-based vegetation detection, boundary extraction, and automatic determination of principal directions. In their follow-on work, they mainly focused on the modeling step in the workflow and extended the classic dual contouring of hermite data [14] to a 2.5D dual-contouring technique [15]. They later improved the approach by adding topology control [16]. This approach produces building models with arbitrary rooftop shapes.

Lafarge *et al.* [17], [18] presented a novel and robust method for modeling cities from point clouds. The algorithm is able to construct simultaneously buildings, trees and terrains. They used LiDAR point clouds for experiments but claiming it is not restricted to LiDAR data inputs.

In consideration of the combination of LiDAR and geometrically uncorrected image data for urban modeling, some work has been done on automatic registration of aerial images with LiDAR data. Ding *et al.* [19] and Lu *et al.* [20] both proposed registration method based on 3D feature detection and matching. Mastin *et al.* [21] introduced a novel idea for utilizing mutual information between the LiDAR and the 2D imagery, which analyzes the statistical dependency in scenes. This information merger has advantages, however, finding correspondences between these two different types of data automatically is often problematic.

III. Methodology Overview

In this paper, a fully automatic method is proposed (see Fig. 1) that processes airborne LiDAR point cloud data for the purpose of building detection and modeling. The LiDAR data is a group of unorganized discrete points in which each individual point has an x, y, and z value, plus the intensity value that represents the reflective properties of surface encountered. Our algorithm requires only LiDAR data as input with no prerequisite to rasterize this data. The output of our algorithm is a group of watertight mesh models that could be used for various applications. We follow a standardized pipeline and introduce novel methodologies to improve accuracy and efficiency. The steps in the workflow pipeline are:

1) **Preparation**: The input point cloud is first passed through a noise filter (statistical outlier removal [22]) in order to obtain a less noisy data set. The point normals and curvatures of all input points are estimated based on basic eigenanalysis.
2) **Scene Classification**: A graph cuts based optimization algorithm is introduced which examines the local distributions of point normals. These distributions exhibit vastly different behavior with a strong dependence on which landcover category (vegetation of non-vegetation) a set of points belongs to. A hierarchical Euclidean clustering

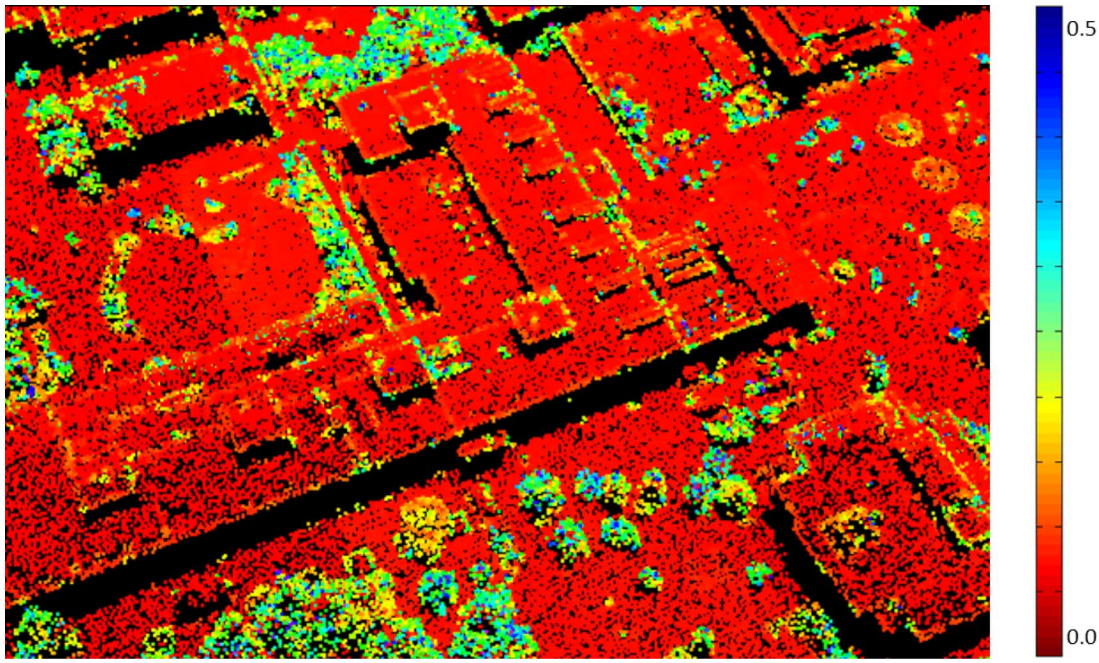


Fig. 2. 3D color plot according to the value of λ_2^n: reddish color indicates smaller values; greenish color indicates larger values.

method is then proposed to initially extract the terrain from the tree-excluded scene, followed by the extraction of rooftop patches. After this step, each building footprint is successfully detected (see the second block of the flowchart in Fig. 1).

3) **Rooftop Features Detection**: In order to describe each individual building rooftop with the best spatial detail possible, a region growing, segmentation method is developed with a smoothness constraint and curvature consistency to detect every significant feature on the rooftop (see the third block of the flowchart in Fig. 1).
4) **Boundary Production**: Points on the boundaries of all detected parts on the rooftop are generated by applying rectilinear fitting see the fourth block of the flowchart in Fig. 1.
5) **Building modeling**: A robust 2.5D dual contouring method is utilized to generate facetized, watertight building models see the last block of the flowchart in Fig. 1.

IV. Classification

The primary goal of classification in this research is to divide the scene into three categories: vegetation, building footprints, and the terrain. The classification is conducted in two separate steps. The first step is to filter out the vegetation areas based on exploring the local implicit surface property of the point cloud. In this step, the point normal is the only used feature which provides the simplicity of the algorithm. The second step is to extract the terrain and roof footprints from the tree-excluded scene obtained in the first step. These two steps are tightly connected but based on independent approaches. The performance of terrain and roof footprints extraction relies on the vegetation detection and removal in the first step.

A. Examine Local Distribution of Normals

In order to explore the properties of the local distribution of normals, the normal of each point needs to be estimated. The estimation is calculated within a local neighborhood of the query point. This local neighborhood can be defined by the spatial relationships of the points in terms of 3D point cloud processing. The points in the neighborhood should be sufficient to represent a small surface patch for feature analysis. Let $p \in P$ be a sample point in the original point cloud, $N_p = \{q | q \in P, d(p,q) < r\}$ be the set of points within a radius range of point p, and $\bar{p}$ is the centroid of all points in N_p, r is the search radius which determines the size of the neighborhood. A common solution for finding the point normal is to solve for the primary eigenvectors of the covariance matrix of points in this neighborhood. The covariance matrix is defined as

$$\mathcal{C}_p = \frac{1}{|N_p|} \sum_{q \in N_p} (q - \bar{p})(q - \bar{p})^T \tag{1}$$

The three eigenvalues $(\lambda_1 < \lambda_2 < \lambda_3)$ can be computed and the eigenvector corresponding to the smallest eigenvalue, λ_1, is used as an estimate for the normal vector associated with point p.

A second covariance analysis (2) is conducted on the collection of estimated normals in small neighborhoods, N_p^n. The covariance matrix in this case is computed as

$$\mathcal{C}_p^n = \frac{1}{|N_p^n|} \sum_{q \in N_p^n} n_q \cdot n_q^T \tag{2}$$

The three eigenvalues are arranged in an increasing order $(\lambda_1^n < \lambda_2^n < \lambda_3^n)$. λ_2^n has been shown to quantitatively indicates the maximum variation of normals on the Gaussian sphere [23], and it is the value that will be considered in the next section.

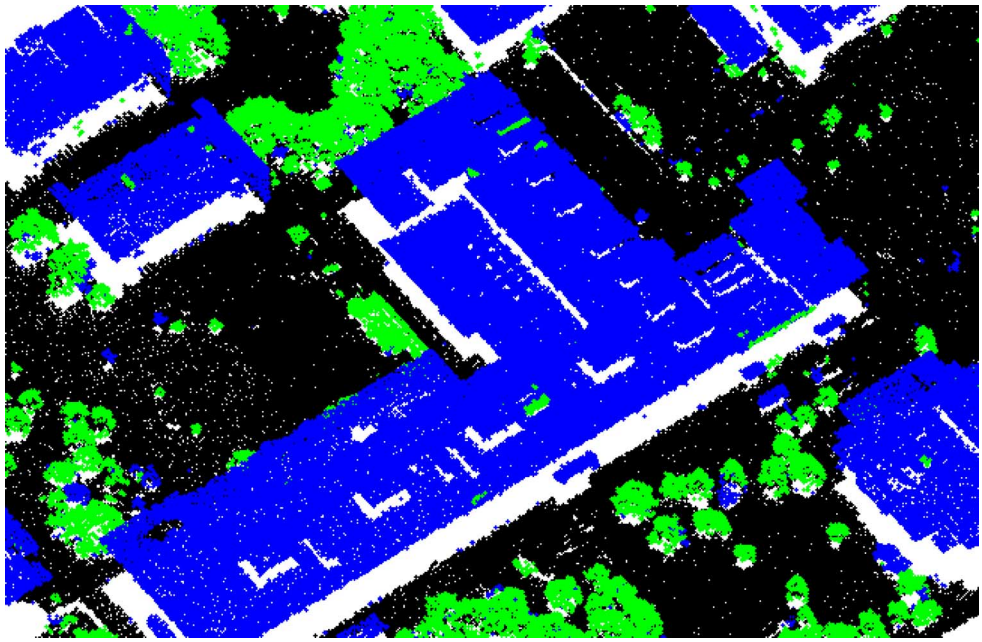


Fig. 3. The example scene classification result (blue: buildings; green: trees; black: terrain). The trees are detected by the graph cuts optimization algorithm introduced in Section IV.B. The terrain and the building footprints are separated by the Euclidean clustering method introduced in Section IV.C.

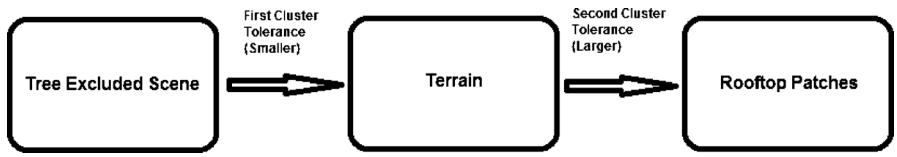


Fig. 4. Flow of hierarchical Euclidean clustering process.

B. Graph Cuts Optimization

In computer vision, a larger number of tasks address the problem of assigning a label to each pixel in a 2D image. For 3D point cloud data, we would like to conduct the same task and assign each three-dimensional point a label. A graph cut algorithm with α-expansion move [24], [25] will be used for point classification problem at hand.

The minimum graph cut algorithm is a powerful tool initially designed for binary optimization. The classification problem being considered here, at this stage, can be nicely regarded as a binary discrimination problem. We categorize the whole scene content into two types which can be noted as $\mathcal{L} = \{L_1, L_2\} = \{\text{trees},\ \text{nontrees}\}$.

The energy function is given by (3). The first term is referred to as the data term (4) while the second term is called the smoothness term (5). ρ_1 and ρ_2 are critical weighting coefficients which indicate the contributions from the two terms in the energy function. l is the label that is chosen for a point.

$$E(l) = \rho_1 E_{\text{data}}(l) + \rho_2 E_{\text{smooth}}(l) \tag{3}$$

$$E_{\text{data}}(l) = \sum_{p \in P} D_p(l_p) \tag{4}$$

$$E_{\text{smooth}}(l) = \sum_{\{p,q\} \in N} V_{p,q}(l_p, l_q) \tag{5}$$

In this approach, each point in the point cloud represents a node of the graph. Each point is connected by its four nearest neighbors. These connections are edges in the graph and can be weighted proportionally by distance. In the context of this problem, the data term presents a penalty if a point with a small λ_2^n value (the variation of local distribution of normals) is encountered and is labeled as a tree, and vice versa. The smoothness term becomes a penalty if the connected points are labeled differently. The detailed terms in (4) and (5) are

$$D_p(l_p) = \begin{cases} 1 & \text{if } (\lambda_2^n(p) > T)\,\&(l_p = L_2) \text{ or} \\ & (\lambda_2^n(p) < T)\,\&(l_p = L_1) \\ 0 & \text{if } (\lambda_2^n(p) > T)\,\&(l_p = L_1) \text{ or} \\ & (\lambda_2^n(p) < T)\,\&(l_p = L_2) \end{cases} \tag{6}$$

$$V_{p,q}(l_p, l_q) = \begin{cases} 0 & \text{if } l_p = l_q \\ 1 & \text{if } l_p \neq l_q \end{cases} \tag{7}$$

where p and q denote two points in the point cloud and l_p is the current label of point p. T is a threshold which is specifically tunable. The α-expansion move is used to efficiently optimize the labeling of all points with respect to two pre-defined categories L_1 and L_2.

For the test case used, the ratio between ρ_1 and ρ_2 is chosen to be 1:10, and the classification result is shown in Fig. 3. In another example of result shown in Fig. 5, the top image shows a color coded DSM which clearly shows what areas are trees with the help of visible wavelength imagery of the same study

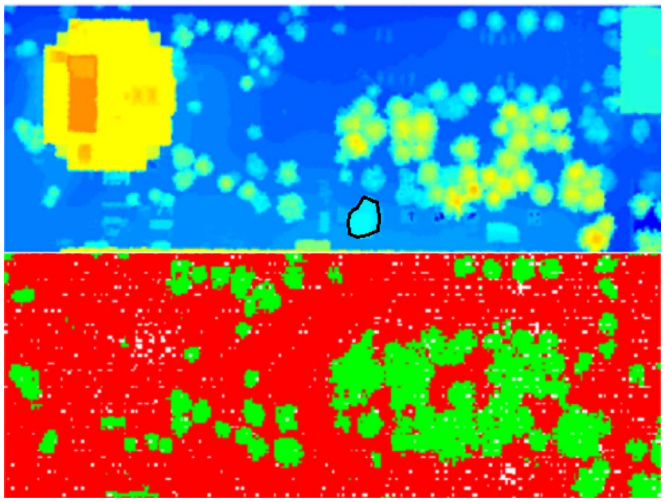


Fig. 5. Visual inspection and comparison on the tree classification accuracy. The black polygon indicates an area that is missed by the proposed detecting algorithm.

site, and the bottom image shows the vegetation detection result (green points are detected trees). It is noticeable that there is a miss of detection that is outlined by a black polygon. In this scene, there are about 15680 points which should belong to the tree category. However only about 15012 points are successfully identified. Therefore, by visual inspection and comparison, the accuracy is estimated to be approximately 95.7%. Compared to the SVM-based vegetation detection algorithm proposed in the work of Zhou *et al.* [13], our classification method is faster, simpler without the loss of accuracy. The most important advantage is we don't involve training data, and we currently consider only one geometry property which is derived from the normal vector estimation. The threshold T used in the data term can be derived from the input data based on the histogram of λ_2^n. There is no need to involve another set of training data to determine it.

C. Terrain and Rooftop Patches Extraction

Graph cuts optimization helped to remove vegetation from the scene. Our major working targets are man-made structures (buildings specifically). So, the next step is to detect terrain and extract all possible rooftop patches from the scene which has already had vegetation excluded.

As long as various targets in the scene are spatially separated from one another, it is reasonable to apply the Euclidean clustering introduced in Rusu's work [22] to group and identify points that make up these targets. In this approach, a strategy referred to as hierarchical Euclidean clustering (Fig. 4) is proposed. Two consecutive clustering passes, utilizing different tolerances which determine the searching radii in the neighborhood, are conducted. Generally speaking, the terrain presents the largest areal coverage in many urban scenes. The first step attempts to separate this largest area from the rest of the scene elements (the buildings). Since airborne LiDAR data is collected above the ground, there are very few points that fall on the sides of the buildings compared to the points that are on rooftops. This fact guarantees the terrain portion of the scene will have very little contact with the buildings, or other man-made objects, and can be successfully isolated and removed based on Euclidean clustering with a reasonable tolerance value (the first step in Fig. 4). Fig. 3 depicts the terrain in black. Once the terrain has been extracted, the remaining features, primarily buildings, can be also easily clustered by using a larger tolerance value (larger searching radius) than the one used in the first clustering pass (the second step in Fig. 4). In our test case, the first tolerance value is set to be 1m, and the second tolerance value is set to be 3 m.

During the clustering process, the number of clusters is not specified in advance. However, after the process is finished, a number of clusters are abandoned due to small membership. This is not looked at as a deficiency in the approach since these small collections of points, very likely, do not represent meaningful structures. The rooftop patches extracted from the tree-exclusive scene can be seen in Fig. 6 (the terrain is shown in blue).

One important concern is that it might not be true that the terrain should be the largest regions detected by the algorithm if only a limited region of some extremely dense urban scene, such as the New York city or the city of San Francisco, is being examined. However, these special cases won't limit the application of this algorithm. The ground portion will still be detected as a self-connected region which may not be the largest one. The algorithm could further examine the average elevation value in this region and then make a decision on if it is the terrain or not. Another important issue that needs to be clarified is the clustering method on terrain extraction makes no assumption that the terrain should be flat on the scale of the entire point cloud. The algorithm mainly looks at the connectivity of the points that could be presumably grouped together, which imposes no hard constraint on the elevation variation across the whole scene. As matter of fact, the terrain in the testing scene is not a flat ground surface. The elevation changes from about 150 meters to about 170 meters.

V. Building Description

A. Planar Features on Rooftop Detection

After obtaining all of the desired building rooftop patches, each building footprint (cluster) can be processed making use of some powerful computing resources built on parallel systems. At this stage, it is possible to build a model from each individual point set, constructing a single building from each collection. However, in order to achieve models containing as much fine detail as possible, it is necessary to identify the many significant features that exist on rooftops and represent them with as much detail as possible. This refinement process represents another region growing based segmentation problem. Rabbani *et al.* [26] proposed a segmentation method using a smoothness constraint which avoids calculating properties like curvatures. Instead, they calculated the residual value obtained by plane fitting to a small surface area and utilized it as a substitution to the curvature property. Their method was only tested on surfaces of indoor objects in relatively small size not large surfaces like rooftops. The algorithm developed in the approach presented here is very similar. The smoothness constraint is maintained,

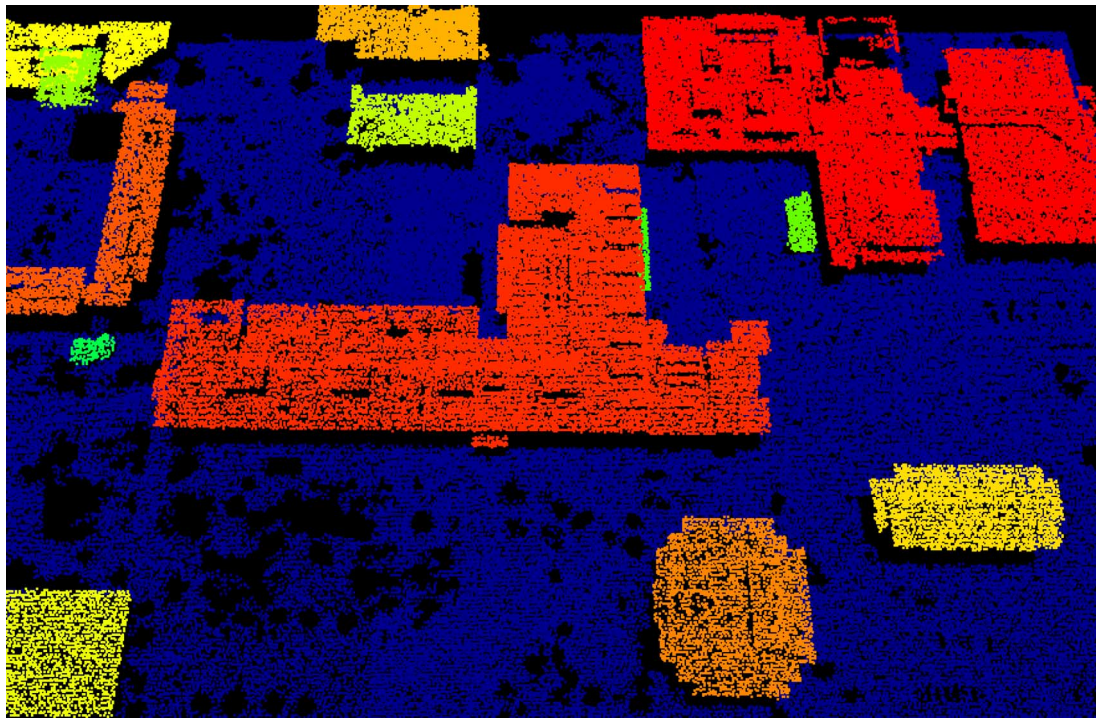


Fig. 6. Terrain and rooftop patches in colors (trees removed).

however, the curvature property is also directly explored. This method works specifically for building rooftops.

Our region growing segmentation process uses the point normals and their curvatures. The curvature estimation is an extension to (1) and is calculated as shown in (8). The process examines the local connectivity and surface smoothness. It first picks a point with the smallest curvature value. Within a small neighborhood of this seed point, it compares the direction of the normal vector of any other point with the normal direction of this seed point. If the directional difference is larger than a predetermined threshold, the point being examined doesn't belong to the group initiated by the seed point, otherwise, it does. In those points which have been grouped together by the seed point, points with curvature values lower than a predetermined threshold are chosen as future seed points. The procedure continues in the same fashion and stops when all points have been visited. During the growing process, the curvature property helps to group points which are supposed to belong to the same region. The algorithm flow is depicted in Fig. 7 in details. For the most part, this approach can successfully isolate the major regions, which consist of a complete rooftop, and assumes thats all the rooftops that are encountered contain only planar surfaces. For each segmented region, RANSAC [27] is applied to fit a virtual plane from the candidate points, and then the points are forced to move on to this estimated plane in order to assign a perfect flatness property to each surface.

$$c = \frac{\lambda_1}{\lambda_1 + \lambda_2 + \lambda_3} \tag{8}$$

Since the normals and curvatures are estimated as closely as possible, yet still are represented only by their best approximation, it is inevitable that over-segmentation will occur due to the nature of the problem. A great number of tiny regions in which there are only a few points, or even only one point, are

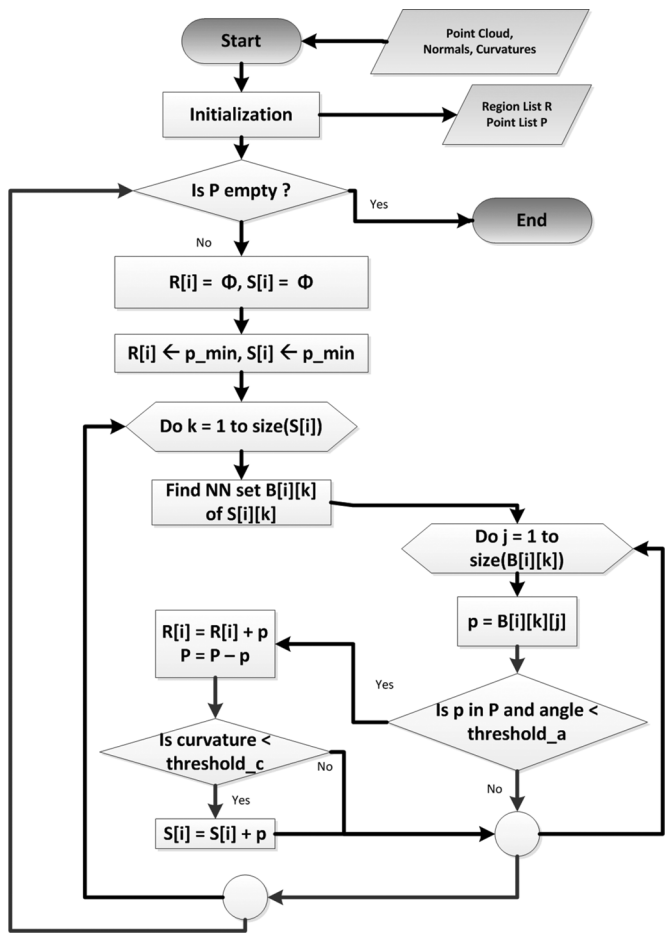


Fig. 7. Region growing using smoothness constraint and curvature consistency.

generated. Another refinement step needs to occur to merge or combine these tiny regions with the major regions to which the

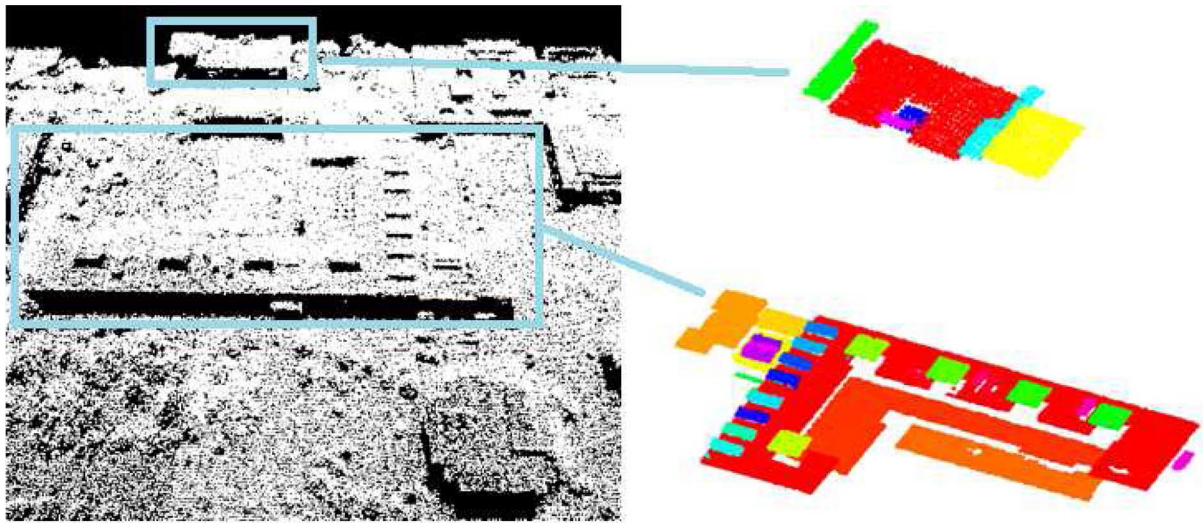


Fig. 8. Each of the two example rooftops is segmented into different parts by our region growing method.

likely belong. For each undetermined point, the distance, d_1, to each estimated plane using RANSAC and the distance, d_2, to each major region are calculated. After normalization, d_1 and d_2 are combined and used as a metric to determine which major region the target point in the tiny region belongs to. Fig. 8 shows an example result. Different segmented regions are assigned different colors.

B. Boundaries Production

The next step in the presented approach is to produce possible boundary points for all features on the rooftops. These points will also be involved in the modeling process later. The shapes of the rooftops in our test scene have great complexities. It is very difficult to directly fit basic geometric primitives to them. Since most common buildings have rectilinear outlines, it is very reasonable to model boundaries of all parts of a complex rooftop under rectilinear constraints. First, the orientation of all features of a building rooftop is fixed to be a dominant orientation, which is currently determined by the prior knowledge. Second, a 2D grid is overlaid on the LiDAR points in the XY plane. The dimension of the grid cell is adjustable. The cells of the 2D grid are marked as being occupied if there are at least a minimum number of points in the cell. This minimum number is determined based on the density of the input point cloud. The boundary of the marked cells now approximately represent the shapes of all parts. The lines are adjusted to be parallel to themselves and closer to the original points for the purpose of achieving a tighter fit. The process is illustrated in Fig. 9.

C. Modeling

The 2.5D dual-contouring approach proposed by Zhou *et al.* [15] is included in our approach for creating watertight building models. Zhou's method deals with points on rooftops only and produces vertical walls connecting these rooftops to the ground. This fits the requirement of this effort. This method extends the classic dual contouring into a 2.5D model. It takes rooftop points, and their estimated normals, as inputs and has no restriction on rooftop shapes. The number of points required for the

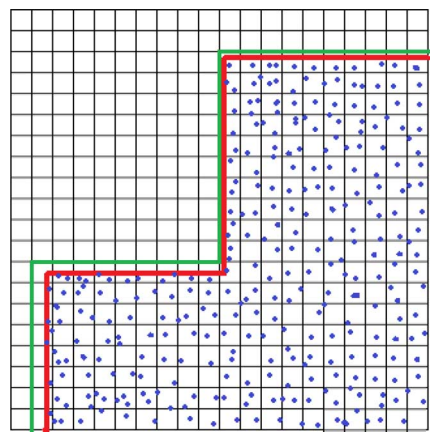
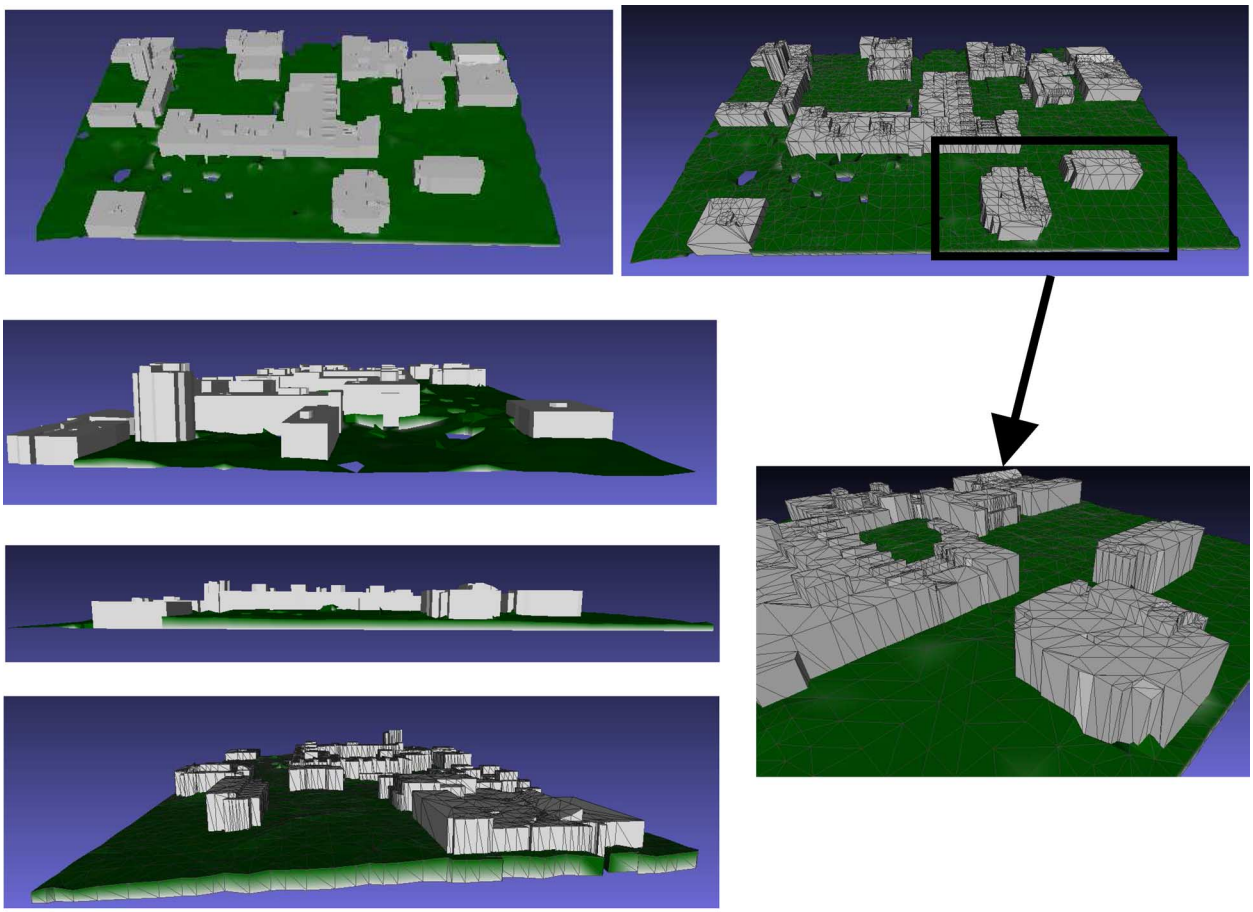


Fig. 9. Finding the rectilinear boundary of a group of points (blue dots). The green line indicates the outline of marked cells. The red line indicates the tightened outline after further fitting.

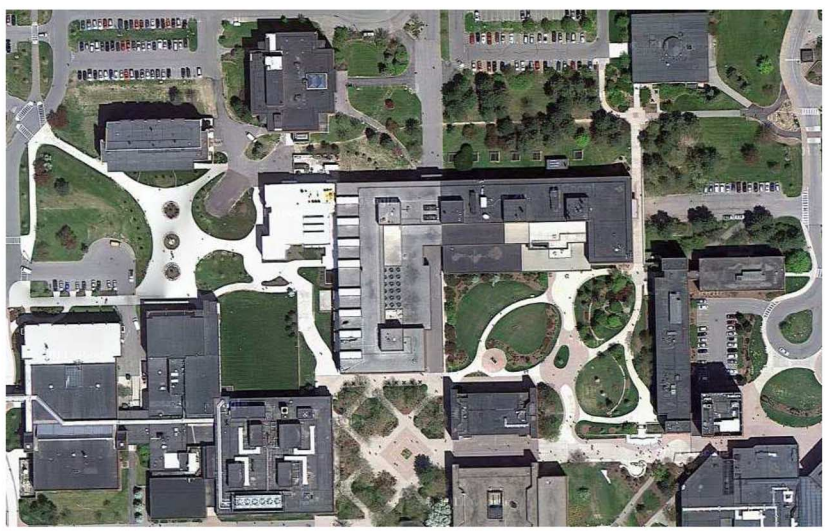
model construction is dramatically reduced, but remains geometrically precise. The topological precision, however, is not guaranteed. This can lead to building rooftop outlines that appear unrealistic and distorted. In follow-on work, [16] proposed a topology control to be added to the 2.5D dual-contouring optimization process. In this current work, every detected feature on the rooftop is treated equally, which achieves the similar goal of maintaining the topology of the structures.

VI. Experimental Results

The proposed workflow has been tested on several airborne LiDAR data collections carried out in the Greater Rochester, New York, USA area. The wavelength of the LiDAR is about 905nm, and the repetition rate is about 25 Hz. The LiDAR point density is approximately 4 samples/m^2 or less. Processing was conducted on a consumer laptop (Intel Core i7, 12G RAM). Figs. 10 and 11 show the reconstruction results for portions of the data collected over the campus of the Rochester Institute of Technology and the City of Rochester. The terrain, in which there is some elevation variation, is also modeled. All models are represented with small triangular facets, which is not only



(a)



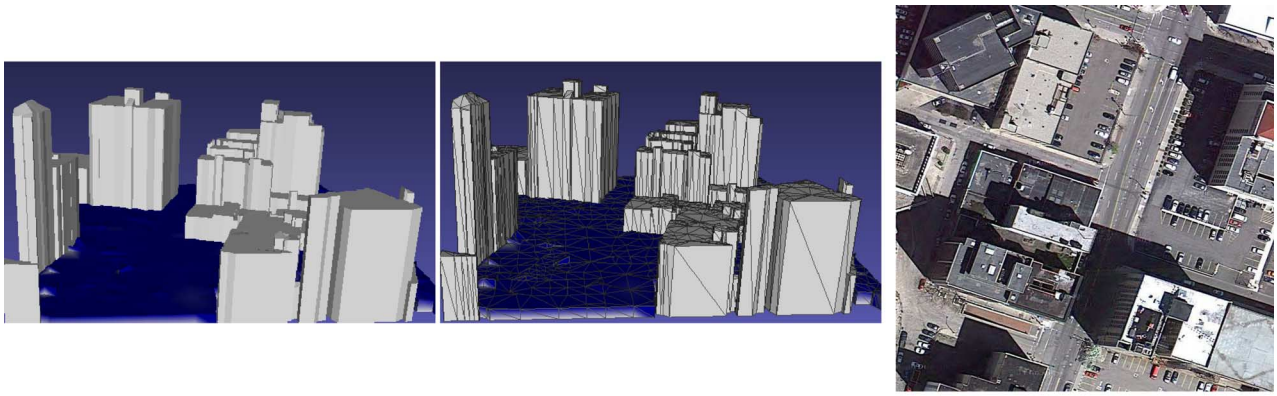
(b)

Fig. 10. (a) Reconstruction of the buildings over the campus of RIT (terrain included). (b) The corresponding optical aerial ortho-photo (courtesy of Google Map).

valuable for visualization, but is also useful for physical simulation in specific applications.

VII. Limitations and Conclusions

There are limitations in the proposed approach. Airborne LiDAR point clouds are the major input to the proposed method which lack of the capability to capture the sides of buildings. This means that each side of a rooftop is connected to the ground by a simple, vertical wall which is obviously not always indicative of the true architectural form. Another concern, which does occur frequently in residential areas, is the partial occlusion of a rooftop by trees. If this occurs, then



(a) (b)

Fig. 11. (a) Reconstruction of the buildings over a part of the city of Rochester (terrain included). (b) The corresponding optical aerial ortho-photo (courtesy of Google Map).

this portion of roof is lost in the LiDAR point cloud, which can result in an incomplete reconstruction. In addition, the 2.5D dual contouring method [15] is a robust algorithm while not responding to our outline refinement ideally. It sometimes introduces a certain degree of distortion to boundaries after facetization. Some inability of the sensor itself also constrains the performance of the proposed approach. For instance, some rooftops are made of or contain areas that are constructed from transparent materials, like glass. The LiDAR scan pulse typically does not provide a detectable return from these surfaces, so these regions may not be able to be reconstructed in the final model.

In general, a method is presented that is fully automatic for 3D building detection and modeling by processing airborne LiDAR point clouds. The building footprints and the terrain are first separated from vegetative areas by applying a graph cuts optimization technique based on the distribution of point normals. When only rooftops and the terrain are left in the scene, a novel hierarchical Euclidean clustering method has been developed to extract rooftop patches and the terrain. For each rooftop patch, a region growing based segmentation method has been presented that detects all significant features on the rooftop. Boundary points are generated under some reasonable assumptions for typical rooftop structures. Finally, the 2.5D dual-contouring method [15] is adopted for modeling process. The models utilize a UTM coordinate system and extracted models are ready to be embedded in to any GIS application.

VIII. Future Work

There are some possible directions for future work. First, the lack of ground truth is a common problem in this field of research, so there are no quantitative error analysis tools available at this time. The authors intend to develop a quality metric for these models in the near future. Second, the number of LiDAR points in a large scene is very large. Dealing with many millions of points efficiently is not an easy task. It is necessary to migrate toward parallel computing techniques for this research and to develop a specifically designed algorithm to process the data. Lastly, a method to accurately determine the principal orientation of a building, or a group of buildings, in a fast, efficient manner is also needed. This will be a focus of future work.

Acknowledgment

The authors would like to thank the United States Department of Energy for their sponsorship of this work.

References

[1] J. Hu, S. You, and U. Neumann, "Approaches to large-scale urban modeling," *IEEE Computer Graphics and Applications*, vol. 23, no. 6, pp. 62–69, Nov.–Dec. 2003.

[2] P. Musialski *et al.*, "A survey of urban reconstruction," in *Eurographics 2012–State of the Art Reports*, 2012, pp. 1–28, Eurographics Assoc..

[3] B. Sirmacek, H. Taubenbock, P. Reinartz, and M. Ehlers, "Performance evaluation for 3-d city model generation of six different dsms from air-and spaceborne sensors," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 5, no. 1, pp. 59–70, 2012.

[4] T. Moons, D. Frère, J. Vandekerckhove, and L. Van Gool, "Automatic modelling and 3D reconstruction of urban house roofs from high resolution aerial imagery," in *Computer Vision, ECCV'98*, 1998, pp. 410–425.

[5] Z. Kim, A. Huertas, and R. Nevatia, "Automatic description of buildings with complex rooftops from multiple images," in *Proc. 2001 IEEE Computer Society Conf., CVPR 2001*, 2001, vol. 2, pp. II-272–II-279.

[6] R. Nevatia and K. Price, "Automatic and interactive modeling of buildings in urban environments from aerial images," in *Proc. 2002 IEEE Int. Conf. Image Processing*, 2002, vol. 3, pp. 525–528.

[7] J. Rau, L. Chen, and G. Wang, "An interactive scheme for building modeling using the split-merge-shape algorithm," *Int. Archives Photogramm. Remote Sens.*, vol. 35, no. B3, pp. 584–589, 2004.

[8] N. Haala and M. Kada, "An update on automatic 3D building reconstruction," *ISPRS J. Photogramm. Remote Sens.*, vol. 65, no. 6, pp. 570–580, 2010.

[9] O. Wang, S. Lodha, and D. Helmbold, "A Bayesian approach to building footprint extraction from aerial lidar data," in *Proc. IEEE 3rd Int. Symp. 3D Data Processing, Visualization, and Transmission*, 2006, pp. 192–199.

[10] V. Verma, R. Kumar, and S. Hsu, "3D building detection and modeling from aerial lidar data," in *Proc. IEEE Computer Society Conf., CVPR 2006*, 2006, vol. 2, pp. 2213–2220.

[11] P. Dorninger and N. Pfeifer, "A comprehensive automated 3D approach for building extraction, reconstruction, and regularization from airborne laser scanning point clouds," *Sensors*, vol. 8, no. 11, pp. 7323–7343, 2008.

[12] C. Poullis and S. You, "Automatic reconstruction of cities from remote sensor data," in *Proc. IEEE Computer Society Conf., CVPR 2009*, 2009, pp. 2775–2782.

[13] Q.-Y. Zhou and U. Neumann, "Fast and extensible building modeling from airborne lidar data," in *Proc. 16th ACM SIGSPATIAL Int. Conf. Advances in Geographic Information Systems, GIS'08*, 2008, pp. 7:1–7:8.

[14] T. Ju, F. Losasso, S. Schaefer, and J. Warren, "Dual contouring of hermite data," *ACM Trans. Graphics*, vol. 21, no. 3, pp. 339–346, 2002.

[15] Q.-Y. Zhou and U. Neumann, "2.5D dual contouring: A robust approach to creating building models from aerial lidar point clouds," in *Computer Vision, ECCV 2010*, 2010, pp. 115–128.
[16] Q.-Y. Zhou and U. Neumann, "2.5D building modeling with topology control," in *Proc. 2011 IEEE Computer Society Conf., CVPR 2011*, 2011, pp. 2489–2496.
[17] F. Lafarge and C. Mallet, "Building large urban environments from unstructured point data," in *Proc. IEEE Int. Conf. Computer Vision*, 2011, pp. 1068–1075.
[18] F. Lafarge and C. Mallet, "Creating large-scale city models from 3D-point clouds: A robust approach with hybrid representation," *Int. J. Computer Vision*, pp. 1–17, 2012.
[19] M. Ding, K. Lyngbaek, and A. Zakhor, "Automatic registration of aerial imagery with untextured 3D lidar models," in *Proc. IEEE Computer Society Conf., CVPR 2008*, 2008, pp. 1–8, IEEE.
[20] L. Wang and U. Neumann, "A robust approach for automatic registration of aerial images with untextured aerial lidar data," in *Proc. 2009 IEEE Computer Society Conf., CVPR 2009*, 2009, pp. 2623–2630.
[21] A. Mastin, J. Kepner, and J. Fisher, "Automatic registration of lidar and optical images of urban scenes," in *Proc. 2009 IEEE Computer Society Conf., CVPR 2009*, 2009, pp. 2639–2646.
[22] R. B. Rusu, "Semantic 3D object maps for everyday manipulation in human living environments," Ph.D. thesis, Technische Universität Muenchen, Munich, Germany, 2009.
[23] M. Pauly*, Point Primitives for Interactive Modeling and Processing of 3D Geometry*. Konstanz, Germany: Hartung-Gorre, 2003.
[24] Y. Boykov, O. Veksler, and R. Zabih, "Fast approximate energy minimization via graph cuts," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 23, no. 11, pp. 1222–1239, 2001.
[25] V. Kolmogorov and R. Zabin, "What energy functions can be minimized via graph cuts?," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 26, no. 2, pp. 147–159, 2004.
[26] T. Rabbani, F. van Den Heuvel, and G. Vosselmann, "Segmentation of point clouds using smoothness constraint," *Int. Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 36, no. 5, pp. 248–253, 2006.
[27] M. Zuliani, "Ransac for Dummies, With Examples Using the RANSAC Toolbox for Matlab and More," 2009 [Online]. Available: http://vision.ece.ucsb.edu/.../RANSAC/docs/RANSAC4Dummies.pdf

Shaohui Sun received the B.S. degree in biomedical engineering in 2006 and the M.S. degree in computer science and technology in 2008 from Sun Yat-sen University, Guangzhou, China. During his Master study, he did an internship at Microsoft China as a software development engineer. He then worked for Tektronix as a software engineer from 2008 to 2010. He is currently working towards the Ph.D. degree in imaging science at the Rochester Institute of Technology, Rochester, NY, USA.

He is a graduate research assistant in the Digital Imaging and Remote Sensing lab. His research interests include remote sensing, photogrammetry, computer vision focusing on three-dimensional urban reconstruction from airborne LiDAR and imagery.

Carl Salvaggio is a Professor of Imaging Science in the Chester F. Carlson Center for Imaging Science at the Rochester Institute of Technology, Rochester, NY, USA. He is a member of the Digital Imaging and Remote Sensing Laboratory teaching and conducting research in digital image processing, remote sensing, and computer science. His particular research interests lie in numerous areas including thermal infrared phenomenology, exploitation, and simulation; design and implementation of novel imaging and ground-based measurement systems; three-dimensional geometry extraction from multi-view imagery and LiDAR data; material optical properties measurement and modeling; and still and motion image processing for various applications.