Complete Residential Urban Area Reconstruction from Dense Aerial LiDAR Point Clouds

Qian-Yi Zhou, Ulrich Neumann

University of Southern California

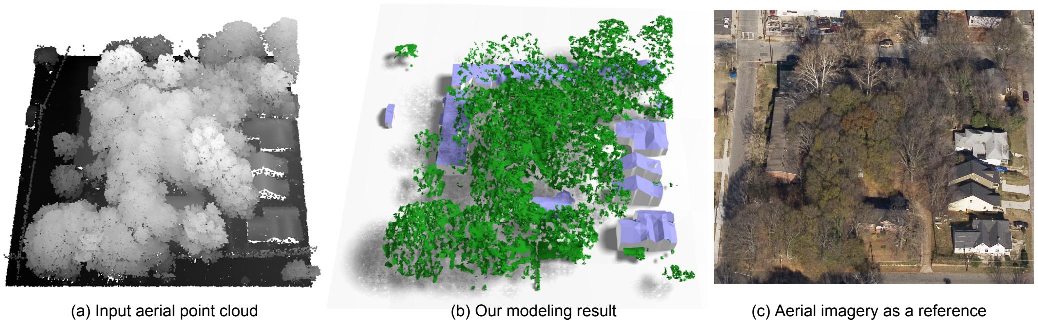
Abstract

We present an automatic system to reconstruct 3D urban models for residential areas from aerial LiDAR scans. The key difference between downtown area modeling and residential area modeling is that the latter usually contains rich vegetation. Thus, we propose a robust classification algorithm that effectively classifies LiDAR points into trees, buildings, and ground. The classification algorithm adopts an energy minimization scheme based on the 2.5D characteristic of building structures: buildings are composed of opaque skyward roof surfaces and vertical walls, making the interior of building structures invisible to laser scans; in contrast, trees do not possess such characteristic and thus point samples can exist underneath tree crowns. Once the point cloud is successfully classified, our system reconstructs buildings and trees respectively, resulting in a hybrid model representing the 3D urban reality of residential areas.

Keywords: urban modeling, LiDAR, residential area, point cloud, classification, 2.5D characteristic, tree modeling

1. Introduction

Urban modeling from aerial LiDAR scans has been an important topic in both computer graphics and computer vision. As researchers mainly focus on downtown areas containing various building structures such as skyscrapers, modern office buildings, stadiums and convention centers; *building reconstruction* is believed to be the core of urban modeling, which has attracted much attention such as [2, 3, 4, 5, 6, 7, 8, 9, 10]. In these efforts, trees are usually considered as an interference to the urban modeling problem, and thus



(a) Input aerial point cloud

(b) Our modeling result

(c) Aerial imagery as a reference

Figure 1: Given (a) a dense aerial LiDAR scan of a residential area (point intensities represent heights), we reconstruct (b) 3D geometry for buildings and trees respectively. (c) Aerial imagery is shown as a reference. This figure was previously published in [1] and is republished with permission by Springer.

are detected and removed from the input by classification in pre-processing. Existing classification algorithms apply heuristics or machine learning approaches on point features including height, intensity, and local geometry information.

However, two new challenges emerge when the urban modeling problem extends to residential areas. First, as shown in Figure 1(a), vegetation is a major component of urban reality in residential areas. An urban modeling method for residential areas should detect and reconstruct both buildings and trees, *e.g.*, as we did in Figure 1(b). The second challenge lies in the classification method: dense LiDAR scans capture the detailed geometry of tree crowns, which may have similar height and local geometry features as rooftops of residential buildings. Figure 2 shows such an example where part of the tree crown shows similar or even better *planarity* than part of the rooftop (see closeups illustrating local points as spheres together with the optimal plane fitted to them). Classification algorithms based on local geometry features may fail and produce significant modeling errors. *E.g.*, Figure 2 right.

To address these two challenges, we present a robust classification method to classify input points into trees, buildings, and ground. Building models and trees are created from these points using a state-of-the-art building reconstruction algorithm [8] and a novel leaf-based tree modeling approach, respectively. The heart of our classification method is a simple, intuitive, but extremely effective measurement. In particular, we observe that residential buildings usually show a strong 2.5D characteristic, *i.e.*, they are composed

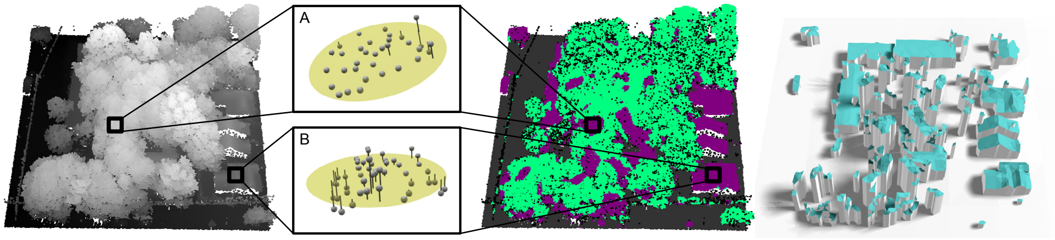


Figure 2: Local geometry features become unreliable when dealing with residential areas with rich vegetation. In closeups of (A) a tree crown region and (B) a rooftop region, points are rendered as spheres while a locally fitted plane is rendered in yellow. Middle: classification results of [7], trees in green, buildings in purple, and ground in dark grey. Right: modeling artifacts are created because of classification errors. Part of this figure was previously published in [1] and is republished with permission by Springer.

of skywards roofs and vertical walls; both are opaque and thus prevent the laser beams from penetrating the building structure. Therefore, there is no point sample inside the building structure. The rooftops (or ground) become the lowest visible surface at a certain x-y position, as illustrated in Figure 3 left. In contrast, trees, composed of branches and leaves, do not have this 2.5D structure. With multiple passes of scanning from different angles, the point cloud captures not only the top surface of the tree crown, but also surfaces inside and underneath the crown, as shown in Figure 3 right.

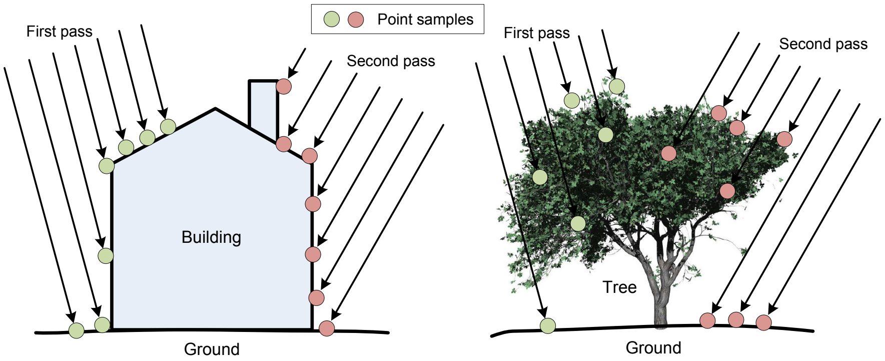


Figure 3: While building structures have a 2.5D characteristic, trees do not possess such property. Dense laser scans may capture surface points under the tree crown (right). This figure was previously published in [1] and is republished with permission by Springer.

Contributions: To the best of our knowledge, we are the first to address the urban modeling problem for residential areas with rich vegetation from aerial LiDAR scans. We specifically list our novelties as follows:

1. We observe the key difference between building structures and trees from the perspective of the 2.5D characteristic. Based on this observation, we propose an effective algorithm to classify trees, building roofs, and ground.
2. We propose a complete system for urban reconstruction of residential areas. A hybrid model containing both 2.5D building models and leaf-based tree models is generated in an automatic and robust manner.

Given that this paper is an extension to the conference paper [1], we list the major improvements as follows. In Section 2, we review the related work from three aspects and explicitly identify the novelties of our method. Section 3 includes more implementation details and an analysis of our classification algorithm demonstrated using Figure 4. In Section 4, concepts from [8] are briefly explained in the footnotes; a discussion regarding occlusion in the building modeling process is given in Section 4.2. We add more experimental results in Section 5 including: qualitative experimental results on two new data sets, statistical tables, and a quantitative comparison based on a novel error measurement. Finally, possible future work is proposed in Section 6.

2. Related Work

We review the related work from three aspects: urban modeling from aerial LiDAR, tree detection in LiDAR, and tree modeling.

2.1. Urban Modeling from Aerial LiDAR

Urban modeling from aerial LiDAR is an important topic that has drawn much attention in both computer graphics and computer vision communities. Recent research work [4, 5, 6, 7] introduces an automatic urban modeling pipeline involving three key steps: *classification* detects and removes trees from the input point cloud; *segmentation* splits individual building roof patches out of the ground; and *building reconstruction* focuses on creating compact and accurate mesh models to represent the geometry of building structures.

Since downtown areas are usually the main target of reconstruction, modern urban modeling methods emphasize on building structures. For instance, Verma *et al.*[6] explore the roof topology graph connecting planar roof patches. Lafarge *et al.*[2] find the optimal configuration of 3D building primitives using a RJMCMC sampler. Matei *et al.*[4] and Poullis and You [5] create building models adapted to Manhattan-World grammars via different approaches. Zebedin *et al.*[11] generate both planar roof patches and surfaces of revolution. Toshev *et al.*[12] propose parse trees as a semantic representation of building structures. Lafarge and Mallet [3] combine primitives and a general mesh representation to achieve hybrid reconstruction. Shen *et al.*[13] focus on building facade segmentation. Zhou and Neumann develop both data-driven modeling approaches [8, 9] and primitive-based method that supports global regularities [10].

The **2.5D characteristic of building models** is first formally observed and defined in [8], as "building structures being composed of detailed roofs and vertical walls connecting roof layers". Many research efforts exploit this characteristic to help building reconstruction either implicitly [4, 5, 6] or explicitly [3, 8, 9, 10].

We extend the emphasis of urban modeling problem from downtown areas to residential areas. As stated in Section 1, we address the challenges in both classification and reconstruction brought by rich vegetation in the residential areas.

2.2. Tree Detection in LiDAR

In urban modeling systems, trees are often recognized as outliers and thus are classified and removed in the first step. Most of the classification algorithms rely on point-wise features including height [3, 14, 15, 12] and its variation [16, 14, 15], intensity [14, 15], and local geometry information such as *planarity* [3, 6, 7], *scatter* [3, 12, 7], and other local geometry features. Heuristics or machine learning algorithms are introduced as classifiers based on the defined feature set. To further identify individual building roof patches, segmentation is either introduced in a post-classification step, or combined with classification in the form of energy minimization such as [3].

Nevertheless, we are the first to introduce the 2.5D characteristic of building structures into the classification problem. We propose a simple, efficient and effective classification algorithm that gains great accuracy in residential areas with rich vegetation.

2.3. Tree Modeling

Tree modeling is a missing part in most of the aerial LiDAR based urban modeling approaches. To the best of our knowledge, Lafarge and Mallet [3] is the only research work which addresses the tree modeling problem by matching simple ellipsoidal template to tree clusters. This method, however, is problematic when dealing with complicated tree structures in residential areas, *e.g.*, Figure 1(a).

Computer graphics and remote sensing communities have made great efforts in modeling trees from ground LiDAR and imagery, such as [17, 18, 19, 20, 21, 22]. A general tree model is broadly adopted in these literatures, composed of skeletal branches and leaves attached to them. Inspired by these efforts, we propose leaf-based tree modeling from aerial LiDAR scans.

3. Point Cloud Classification

Given an aerial LiDAR point cloud of a residential area as input, the objective of classification is to classify points into three categories: trees, buildings, and ground. As mentioned in Section 1 and illustrated in Figure 3, the 2.5D characteristic is the key difference between trees and buildings (or ground). In order to formulate this concept, we discretize the point cloud by embedding it into a uniform 2D grid G. In each grid cell c, the point set $P(c)$ is segmented into multiple *layer fragments* $L(c)$, using local distance-based region growing. Ideally, a layer fragment $l_{building} \in L(c)$ lying on a 2.5D object (rooftop or ground) must have the lowest height among all layer fragments in $L(c)$, because the rooftop (or ground) is always the lowest visible surface to laser beams at a certain x-y position, as analyzed in Section 1. On the other hand, a tree layer fragment l_{tree} can exhibit any height. However, as there is usually a ground or rooftop surface underneath tree samples, l_{tree} is not expected to be the lowest layer fragment in $L(c)$. Therefore, we check all the layer fragments in each cell, assign only the lowest layer fragment as non-trees (rooftop or ground), and classify the rest layer fragments as trees. As shown in Figure 4(b), this novel criterion creates good classification results between trees and non-trees (for clarity, trees are not shown in this figure). From an energy minimization perspective, this 2.5D characteristic criterion can be quantized with a data energy term $E_d(x_l)$ for each $l \in L(c)$ as:

$$E_d(x_l) = \begin{cases} \alpha & \text{if } x_l = \textit{building} \text{ or } \textit{ground}\text{, and } l \text{ is not the lowest in } L(c) \\ \beta & \text{if } x_l = \textit{tree}\text{, and } l \text{ is the lowest layer fragment in } L(c) \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

where x_l is the label of layer fragment l.

To further discriminate building and ground in the energy minimization framework, we introduce *elevation* of layer fragment $e(l)$ defined as the height difference between l and the ground elevation at c's center[1]. Another data energy term $E_g(x_l)$ is defined accordingly:

$$E_g(x_l) = \begin{cases} \gamma \cdot max(1 - \frac{e(l)}{\sigma}, 0) & \text{if } x_l = building \\ \gamma \cdot min(\frac{e(l)}{\sigma}, 1) & \text{if } x_l = ground \\ 0 & \text{if } x_l = tree \end{cases} \tag{2}$$

where σ is the normalization factor. Empirically, $\sigma = 6m$, as suggested in [3].

With a smooth energy $E_s(x_{l_1}, x_{l_2})$ defined over all neighboring layer fragment pairs (*i.e.*, layer fragments belonging to neighboring cells and satisfying certain distance criteria), we build a Markov Random Field which leads to an energy minimization problem over the labeling x of the entire layer fragment set $\mathcal{L}$:

$$E(x) = \sum_{l \in \mathcal{L}} (E_d(x_l) + E_g(x_l)) + \lambda \sum_{(l_1, l_2) \in \mathcal{N}} E_s(x_{l_1}, x_{l_2}) \tag{3}$$

where $\mathcal{N}$ is the set of neighboring layer fragment pairs, and smooth energy $E_s(x_{l_1}, x_{l_2})$ is defined as characteristic function $\mathbf{1}_{x_{l_1} \neq x_{l_2}}$.

With the energy minimization problem being solved using the well-known graph-cut method [23], point labels are determined as the label of the corresponding layer fragment. To further construct roof patches from building points, a region growing algorithm is applied based on certain distance criteria. While large building patches are adopted as rooftops, small patches are considered as outliers and removed henceforth.

Figure 4 demonstrates the entire process of point cloud classification. Input points are first discretized into layer fragments. For illustration purpose, Figure 4(b) shows only the lowest layer fragment in each cell. They faithfully capture the skyward surfaces of 2.5D structures. By solving an energy minimization problem, building and ground layer fragments are detected and shown in Figure 4(c). Because of the smooth energy term, eaves are segmented as part of the building roof, and small clusters of tree layer

[1] Empirically, the ground elevation map is easily estimated by assigning a 20m-by-20m coarse grid, estimating ground height with the lowest point in each cell, and applying linear interpolation across the entire coarse grid. More accurate (and more complicated) ground elevation estimation methods can be found in related literatures [3, 12, 7].

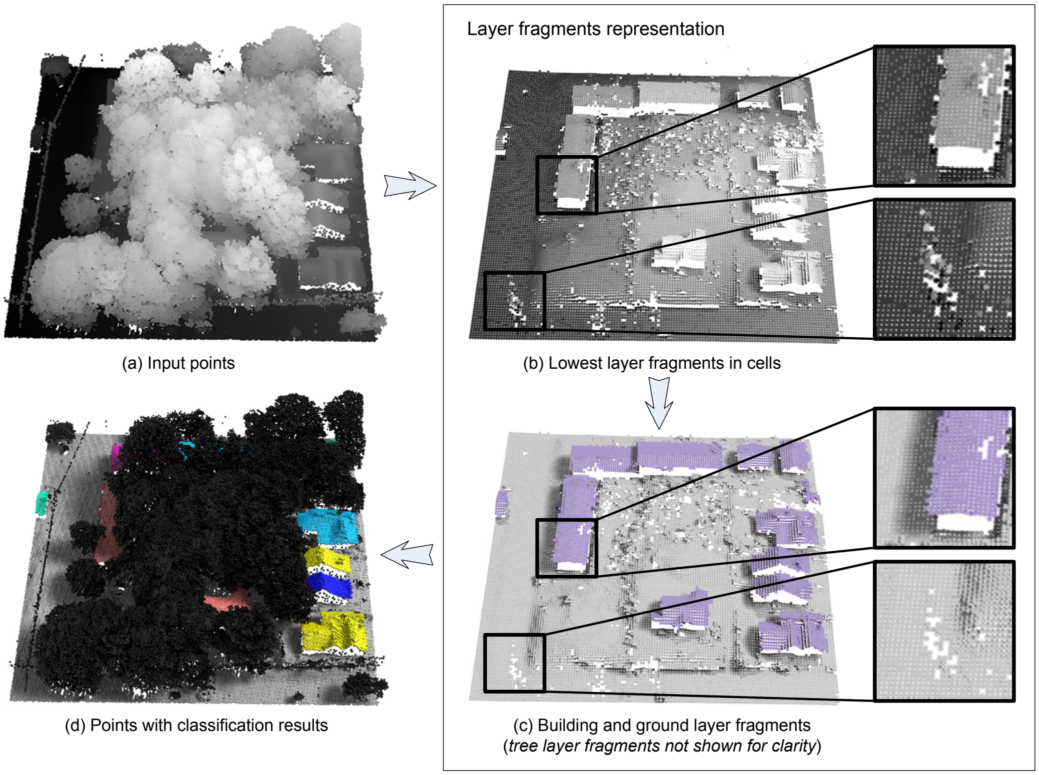

Layer fragments representation
(a) Input points
(b) Lowest layer fragments in cells
(c) Building and ground layer fragments
(*tree layer fragments not shown for clarity*)
(d) Points with classification results

Figure 4: A demonstration of the classification algorithm: (b) the lowest layer fragments faithfully capture the skyward surfaces of 2.5D structures; (c) building and ground layer fragments are rendered in purple and grey respectively; (d) trees and outliers are in black while building roof patches are rendered in bright colors.

fragments with low heights are correctly detected, as illustrated in the close-ups respectively. Finally, the classification result is applied to the point cloud and a region growing algorithm successfully groups roof patches from building points, *i.e.*, Figure 4(d).

Here we highlight the differences between our approach and the classification algorithm in [3]. Although both methods utilize a Markov Random Field, the key features in [3] that classify trees and non-trees are local geometry features (*planarity* and *scatter*). Thus, it may fail for residential areas due to similar local geometry features of trees and non-trees, as demonstrated in Figure 2. On contrast, our classification approach benefits from the powerful 2.5D characteristic criterion, and thus successfully classify points into trees, building and ground, as shown in Figure 4.

4. Modeling of Urban Elements

Based on the successful classification of input points, we introduce different modeling approaches for trees, buildings, and ground respectively.

4.1. Tree Modeling

Modern tree modeling approaches adopt a general tree structure composed of skeletal branches and leaves attached to them. Tree reconstruction usually begins with a branch generation algorithm followed by a leaf modeling approach. However, unlike ground-based laser scans and imagery, aerial LiDAR data captures very few samples on branches, making branch generation a difficult task. Therefore, we choose to directly model tree leaves by fitting surface shapes around tree points having sufficient neighbors.

In particular, for each tree point p with sufficient neighbors, Principal Component Analysis is applied to its neighboring point set $N(p)$ to fit an ellipsoid. Eigenvectors $\mathbf{v}_0, \mathbf{v}_1, \mathbf{v}_2$ and eigenvalues $\lambda_0, \lambda_1, \lambda_2$ of the covariance matrix represent the axes directions and lengths of the ellipsoid respectively. We employ the inscribed octahedron of the ellipsoid to represent the local leaf shape around p. Specifically, an octahedron is created with six vertices located at $\{\mathbf{v}_p \pm s\lambda_0\mathbf{v}_0, \mathbf{v}_p \pm s\lambda_1\mathbf{v}_1, \mathbf{v}_p \pm s\lambda_2\mathbf{v}_2\}$, where $\mathbf{v}_p$ is the location of p and s is a user-given size parameter.

A uniform sampling over the tree point set P_{tree} can be applied to further reduce the scale of the reconstructed models.

4.2. Building Modeling

We adopt 2.5D dual contouring method [8] to create building models from rooftop patches through three steps: (1) sampling 2.5D Hermite data over a uniform 2D grid, (2) estimating a hyper-point[2] in each grid cell, and (3) generating polygons.

The only challenge in applying 2.5D dual contouring to residential area data lies in rooftop holes caused by occlusion. To solve this problem, we add a hole-filling step right after 2.5D Hermite data is sampled from input points. In particular, we scan the entire 2D grid to detect rooftop holes, and solve a Laplace's equation $\nabla^2 z = 0$ to fill these holes, where z represents

[2] A hyper point is a series of 3D points having the same x-y coordinates but different z values.

Data set	Point #	Resolution	Building #	Building triangle #	Building error rate	Octahedron #
Atlanta area #1	5.5M	22.9/m^2	418	55,568	1.1%	52,924
Atlanta area #2	4.0M	18.8/m^2	323	61,492	0.7%	29,151
Denver area	1.0M	6.3/m^2	290	42,942	0.6%	17,054

Table 1: Statistics of the experiments on three different data sets

Data set	Classification	Normal estimation	Building modeling	Tree modeling	Total time
Atlanta area #1	9s	45s	54s	<1s	109s
Atlanta area #2	6s	29s	32s	<1s	68s
Denver area	3s	8s	11s	<1s	23s

Table 2: Execution time of each step in our system

the heights of *surface Hermite samples* at grid corners[3]. Existing surface Hermite samples serve as the boundary condition of the Laplace's equation.

In practice, since trees do not have the 2.5D structure, a point cloud composed of multiple scans from different perspectives can capture most of the rooftop even it is underneath tree crowns. The occlusion is usually insignificant, and is successfully handled by the hole-filling step.

4.3. Ground Modeling

Ground models can be easily created by rasterizing ground points into a DSM (digital surface model). Holes are filled via linear interpolation.

5. Experimental Results

Our system is tested on various data sets. For each data set, we adapt the following parameter configuration with respect to the data resolution. The length of the grid cell r is set to $\frac{3}{\sqrt{d}}$ given d as the point density in m^{-2}, to ensure sufficient samples in each grid cell. In the initial region growing step which determines layer fragments, a pair of points with distance less than r are taken as neighbors. In our classification algorithm, energy function parameters, *i.e.*, $\{\alpha, \beta, \gamma, \lambda\}$ are set to $\{1.0, 2.0, 0.5, 4.0\}$ empirically. This parameter configuration works well for all the data sets we have tested. Once

[3] Surface Hermite data is sampled per grid corner, by intersecting a vertical line and the rooftop surface [8].

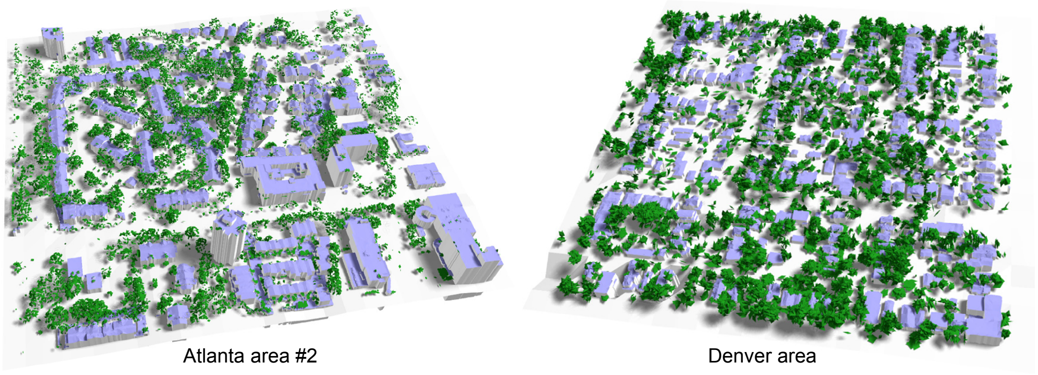


Figure 5: Our residential urban modeling system is tested on multiple data sets from different sources. We robustly reconstruct urban reality despite the variation in data resolution, building patterns, and tree types.

roof points are detected and grouped into roof patches, those roof patches with less than d samples (per m^2 sample number) are considered as outliers and removed henceforth. In tree modeling, octahedron size s is chosen by the user in the interval $[\frac{1}{r}, \frac{3}{r}]$. We show experimental results with $s = \frac{2}{r}$.

Figure 6 shows our urban reconstruction results for a 520m-by-460m residential area in the city of Atlanta. The input contains 5.5M aerial LiDAR points with 22.9/m^2 resolution. Our algorithm reconstructs 56K triangles for building models, and 53K octahedrons as tree leaves, in less than two minutes on a consumer-level laptop. As illustrated in the closeups of Figure 6, our classification algorithm successfully classifies points into trees, ground, and individual building patches (second column). A hybrid urban model is generated by combining 2.5D polygonal building models and leaf-based tree models (third column). Aerial imagery is given in the last column as a reference.

We then test our residential urban modeling system on another two data sets, with data resolution ranging from 6/m^2 to 19/m^2. Visually appealing urban models are reconstructed respectively, despite the variation in point density, building model patterns, and tree types, as shown in Figure 5. To quantitatively evaluate the modeling results, we count the false positives (unexpected results) and false negatives (missing results) of buildings by comparing our results with aerial imagery as a trusted external judgement. In all three experiments, no false negative is found, *i.e.*, all building structures are successfully detected and reconstructed by our system. In addition, false positives exist in the form of small building-like trees and incorrectly classified ground components. We thus calculate the error rate as the ratio of the number of triangles in false positives to the total number of triangles in all building models[4]. Table 1 contains statistics of the three experiments, in which error rates are generally low. Table 2 shows the computation time, measured on a laptop with Intel i-7 CPU 1.60GHz and 6GB memory.

6. Conclusion and Future Work

In this paper, we address the complicated problem of reconstructing urban models for residential areas with rich vegetation. We observe the key difference between buildings and trees in terms of the 2.5D characteristic: while buildings are composed of opaque skyward rooftops and vertical walls, trees allow point samples underneath the crown. This feature enables a powerful classification algorithm based on an energy minimization scheme. By combing classification, building modeling and tree modeling together, our system automatically reconstructs a hybrid model composed of buildings and trees from the aerial LiDAR scan of a residential area. Our experiments demonstrate the effectiveness and efficiency of our system.

Possible future work lies within two major directions. First, besides trees, buildings, and ground, more urban element categories can be explored. For instance, Figure 6(b) top left shows a building under construction (see LiDAR scan and aerial imagery). It is not considered as a complete building structure since it has no roofs. Thus, our current approach classifies it as trees and create leaves for it. Future research may consider it as other category such as *wall* and model it with different manners. The second future research is to integrate aerial LiDAR with other sources such as aerial imagery, ground LiDAR, and street view images. More realistic model can be created by combining both geometry and visual information from various data sources.

[4] In residential areas, buildings and trees have similar local geometry properties, and are spatially close to each other. It is very difficult even for a human being to label ground truth classification on the point cloud. Thus, it is impractical to make traditional quality measurements such as point-based error rate. We adopt the building error rate as a simple and intuitive measurement for our application.

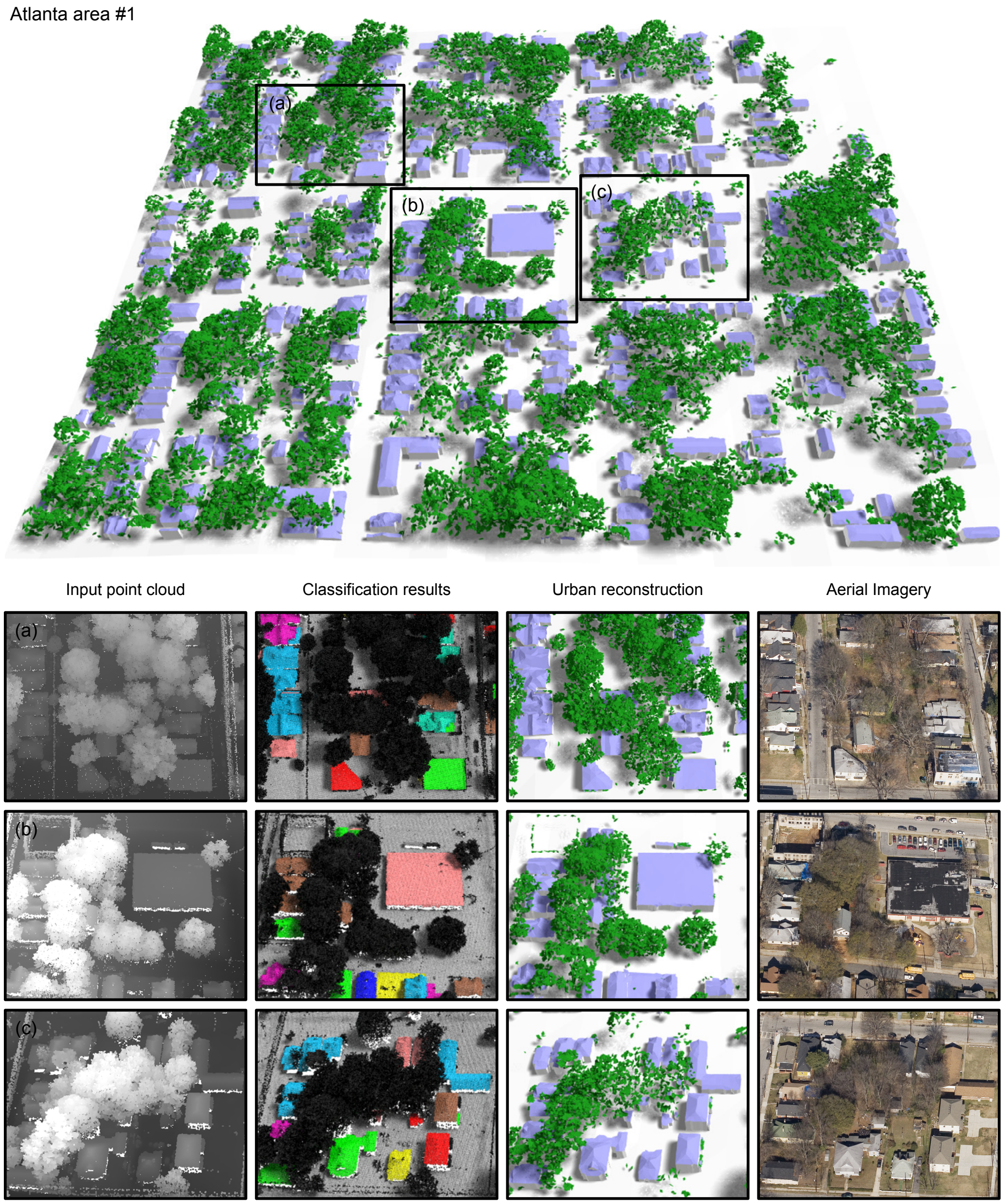


Figure 6: Urban models reconstructed from 5.5M aerial LiDAR points for a residential area in the city of Atlanta. Part of this figure was previously published in [1] and is republished with permission by Springer.

7. Acknowledgement

We thank Airborne 1 Corp. and Sanborn Corp. for providing data. We are grateful for support from Radek Grzeszczuk and Thommen Korah at Nokia Research Center in Palo Alto. We thank Na Chen and anonymous reviewers for their valuable comments.

References

[1] Q.-Y. Zhou, U. Neumann, Modeling residential urban areas from dense aerial lidar point clouds, in: Computational Visual Media Conference, 2012. 2, 3, 4, 13

[2] F. Lafarge, X. Descombes, J. Zerubia, M. Pierrot-Deseilligny, Building reconstruction from a single dem, in: CVPR, 2008. 1, 5

[3] F. Lafarge, C. Mallet, Building large urban environments from unstructured point data, in: ICCV, 2011. 1, 5, 7, 8

[4] B. Matei, H. Sawhney, S. Samarasekera, J. Kim, R. Kumar, Building segmentation for densely built urban regions using aerial lidar data, in: CVPR, 2008. 1, 4, 5

[5] C. Poullis, S. You, Automatic reconstruction of cities from remote sensor data, in: CVPR, 2009. 1, 4, 5

[6] V. Verma, R. Kumar, S. Hsu, 3d building detection and modeling from aerial lidar data, in: CVPR, 2006. 1, 4, 5

[7] Q.-Y. Zhou, U. Neumann, A streaming framework for seamless building reconstruction from large-scale aerial lidar data, in: CVPR, 2009. 1, 3, 4, 5, 7

[8] Q.-Y. Zhou, U. Neumann, 2.5d dual contouring: A robust approach to creating building models from aerial lidar point clouds, in: ECCV, 2010. 1, 2, 4, 5, 9, 10

[9] Q.-Y. Zhou, U. Neumann, 2.5d building modeling with topology control, in: CVPR, 2011. 1, 5

[10] Q.-Y. Zhou, U. Neumann, 2.5d building modeling by discovering global regularities, in: CVPR, 2012. 1, 5

[11] L. Zebedin, J. Bauer, K. Karner, H. Bischof, Fusion of feature- and area-based information for urban buildings modeling from aerial imagery, in: ECCV, 2008. 5

[12] A. Toshev, P. Mordohai, B. Taskar, Detecting and parsing architecture at city scale from range data, in: CVPR, 2010. 5, 7

[13] C.-H. Shen, S.-S. Huang, H. Fu, S.-M. Hu, Adaptive partitioning of urban facades, in: SIGGRAPH ASIA, 2011. 5

[14] S. K. Lodha, D. M. Fitzpatrick, D. P. Helmbold, Aerial lidar data classification using adaboost, in: 3DIM, 2007. 5

[15] J. Secord, A. Zakhor, Tree detection in urban regions using aerial lidar and image data, IEEE Geoscience and Remote Sensing Letters. 5

[16] G. Chen, A. Zakhor, 2d tree detection in large urban landscapes using aerial lidar data, IEEE ICIP. 5

[17] J.-F. Côté, J.-L. Widlowski, R. A. Fournier, M. M. Verstraete, The structural and radiative consistency of three-dimensional tree reconstructions from terrestrial lidar, Remote Sensing of Environment. 6

[18] Y. Livny, S. Pirk, Z. Cheng, F. Yan, O. Deussen, D. Cohen-Or, B. Chen, Texture-lobes for tree modelling, in: ACM SIGGRAPH, 2011. 6

[19] B. Neubert, T. Franken, O. Deussen, Approximate image-based tree-modeling using particle flows, in: ACM SIGGRAPH, 2007. 6

[20] P. Tan, T. Fang, J. Xiao, P. Zhao, L. Quan, Single image tree modeling, in: ACM SIGGRAPH Asia, 2008. 6

[21] P. Tan, G. Zeng, J. Wang, S. B. Kang, L. Quan, Image-based tree modeling, in: ACM SIGGRAPH, 2007. 6

[22] H. Xu, N. Gossett, B. Chen, Knowledge and heuristic-based modeling of laser-scanned trees, ACM Trans. Graph. 6

[23] Y. Boykov, O. Veksler, R. Zabih, Fast approximate energy minimization via graph cuts, IEEE PAMI. 7