3D Building Detection and Modeling from Aerial LIDAR Data

Vivek Verma
Sarnoff Corporation
vverma@sarnoff.com

Rakesh Kumar
Sarnoff Corporation
rkumar@sarnoff.com

Stephen Hsu
Canesta, Inc.
shsu@canesta.com

Abstract

This paper presents a method to detect and construct a 3D geometric model of an urban area with complex buildings using aerial LIDAR (Light Detection and Ranging) data. The LIDAR data collected from a nadir direction is a point cloud containing surface samples of not only the building roofs and terrain but also undesirable clutter from trees, cars, etc. The main contribution of this work is the automatic recognition and estimation of simple parametric shapes that can be combined to model very complex buildings from aerial LIDAR data. The main components of the detection and modeling algorithms are (i) Segmentation of roof and terrain points. (ii) Roof topology Inference. We introduce the concept of a roof-topology graph to represent the relationships between the various planar patches of a complex roof structure. (iii) Parametric roof composition. Simple parametric roof shapes that can be combined to create a complex roof structure of a building are recognized by searching for sub-graphs in its roof-topology graph. (iv) Terrain Modeling. The terrain is identified and modeled as a triangulated mesh. Finally, we provide experimental results that demonstrate the validity of our approach for rapid and automatic building detection and geometric modeling with real LIDAR data. We are able to model cities and other urban areas at the rate of about 10 minutes per sq. mile on a low-end PC.

1. Introduction

Reconstruction of building geometry in an urban environment has many useful applications such as urban planning, cartography, surveillance, wireless network planning, entertainment, virtual tourism, geo-spatial web browsing, etc. Building detection and construction is traditionally done interactively using multiple images, and the field has been well studied by researchers in the computer vision and photogrammetry communities.

LIDAR (Light Detection and Ranging) has gained popularity as a way to quickly collect 3D information about an urban site. LIDAR scanning of a site produces dense, unorganized points that require further processing to identify buildings, trees, and bare ground. The emergence of laser range finders and position sensors would seem to neatly bypass the fundamental difficulties of automatically reconstructing 3D shape and camera poses from 2D image collections. Yet, rapid and highly automated construction of accurate and compact models is an unfulfilled goal. This is because LIDAR data poses several challenges in building detection and construction – limited resolution of the samples near surface edges, presence of noise due to errors in GPS/INS, registration errors, and poor reflectivity properties of some surfaces. Despite these limitations, LIDAR is an excellent sensor to directly measure the depth of objects.

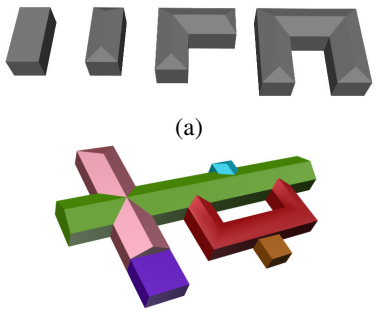
We present a fully automatic approach to creating 3D geometric models of the buildings and terrain from LIDAR data. Our objective is to create compact, watertight geometric models of buildings that fit as closely as possible to the original LIDAR points using minimum number of triangles, as opposed to a dense polygonal mesh. We model the class of buildings that can be constructed by combining several simpler prismatic models. A complex roof shape is automatically decomposed into multiple simpler geometric primitives. Finally, our automatic approach allows building 3D models of cities at an extremely rapid rate. We have achieved modeling speed of about 10 minutes per sq. mile of LIDAR data for urban areas.

2. Overview and Related Work

Given the LIDAR data for an urban area, our objective is to automatically recognize and model complex building roofs as a combination of simple, prismatic parametric shapes rather than general 3D polygonal meshes. In particular, our goals are to automatically segment roofs and terrain from aerial LIDAR and to infer the shape of complex buildings as a combination of simpler parametric shapes.

In this work we make no assumption about prior knowledge of building types. In fact, we automatically infer the roof types directly from LIDAR. We model the class of buildings that can be created using a union of simpler parametric roof types shown in Figures1-(a). For example, the complex roof shaped building in Figure 1-(b) is shown to

Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)
0-7695-2597-0/06 $20.00 © 2006 IEEE



(a)

(b)

Figure 1. (a) Simple parametric shapes that can be used to model complex roof types. (b) A complex roof modeled as a combination of the simple roof types from (a). Each simple sub-roof is colored differently.

be constructed by combining the simple shapes shown in Figure 1-(a).

The LIDAR sensor "sees" only the roofs of the buildings and information about the geometry of the sides of the buildings is not available. Since we are estimating building shapes using only aerial LIDAR, it is not possible to determine the geometric details on the sides of buildings. Our approach is to model the geometry of building sides by extruding the roof outlines down to a ground level (inferred from the LIDAR data itself). One can also acquire LIDAR data using a ground-based vehicle to accurately capture the sides of the buildings. Such an approach is taken by Früh and Zakhor [7]. However, their method is limited because they fit a triangle mesh to the LIDAR points and do not simplify the geometry into parametric models. Hence their 3D models are represented by a soup of polygons that are very hard to edit and inefficient to render. In addition, their 3D models can be quite noisy because the edges of buildings are not straight, perfectly flat walls are not composed of a single plane, etc. These models also lack water-tightness. In contrast, the 3D models generated by our method are watertight, compact, and can be edited by varying a few parameters for each building.

The benefit of representing complex buildings as a combination of parametric shapes is that each building can be described by only a few parameters. For example, a gable roof building can be described by at most 5 parameters (length, width, height, and two slopes). Another advantage in representing buildings as parametric shapes rather than a "soup" of polygons is that they are easier to edit. A building roof or part of a complex building roof that cannot be modeled as one of the parametric shapes shown in Figure 1-(a) is modeled either as a flat or single-sloped roof shape with a rectilinear base. Our system is not limited to recognizing only the shapes shown in Figure 1 because other parametric shapes can be added by a user.

3D building model construction from images is usually posed as the problem of estimating the 3D geometry of building structures from several photographs of the buildings from multiple views. Debevec's Facade system [5] exploits user-selected edge correspondences across multiple images to build a model from 3D primitives and then verifies its accuracy by projecting the model back to the original images. Some methods combine multiple calibrated aerial images, along with some user-guidance, to automatically create 3D models [10]. These methods do not scale to large area modeling because of the manual intervention required. Baillard *et al*. [3] skip the manual calibration step by automatically detecting and matching line segments over multiple aerial images and computing piecewise planar patches that define the building geometry.

Building detection and construction from LIDAR data has also been studied by several researchers. Most methods start by converting the LIDAR point cloud to a depth image [2, 11, 12] and then use well known image segmentation techniques to detect buildings as rectilinear shapes. Other methods assume that a detailed ground plan of the buildings is available and hence detect the height and slope of building structures using LIDAR [8]. The algorithm proposed by You *et al*. [12], fits user selected geometric primitives such as prisms, boxes, cylinders, and super-quadrics, to the LIDAR data. Ahlberg *et al*. [1] and Elaksher *et al*. [6] have developed very similar, completely automatic, building construction methods. Their methods convert the LIDAR point cloud into a depth image and detect edges of a building's roof using a 2D Hough transform. However, their method is very slow and in general produces arbitrary polygonal meshes. These methods use only aerial LIDAR to reconstruct building geometry. Früh and Zakhor [7] combine LIDAR data collected from both aerial and ground-based platforms to create detailed 3D city models in the form of large polygonal meshes. Hovever, these meshes are not guaranteed to be free from cracks. Hu *et al*. [9] have published an extensive survey on the various approaches to extract buildings using imagery and LIDAR data.

In contrast to the previous work, we directly estimate the geometry of the buildings by automatically recognizing simple parametric shapes that can be combined to construct a given complex roof shape.

The rest of the paper is organized as follows: Section 3 presents the overall approach for building recognition and modeling using aerial LIDAR point clouds, Section 4 discusses details of the segmentation of the LIDAR point cloud into roof and terrain points. Section 5 describes the algorithm used to recognize the simple parametric shapes to model a complex roof shape, Section 6 focuses on the 3D building and terrain geometry fitting. We present the experimental results in Section 7 and conclude in Section 8.

3. Approach

Our approach relies on the key observation that most complex roof structures can be modeled as a combination of much simpler roof structures. In Figure 1 we illustrate a few simple parametric geometric primitives that can be combined to model a large number of complex roof types.

There are four major components of our algorithm: roof point segmentation, roof topology inference, parametric shape fitting, and terrain shape estimation.

Roof point segmentation consists of two major steps – removing non-flat points and identification of roof and terrain points. Points whose local neighborhood does not define a plane are considered to be non-flat and typically belong to structures such as trees, fences, poles, etc. An efficient 3D connected components algorithm is used to identify large continuous regions. The topology inference stage consists of creating a topology graph, which we define to be the dual of the roof structure such that each planar surface of the roof is represented by a vertex. Two vertices of the topology graph are connected by an edge if and only if the planar roof surfaces corresponding to the two vertices share an edge. Individual planar roof segments are identified by fitting polygonal patches to the segmented roof points. The parametric shape fitting algorithm searches the topology graph for sub-graphs corresponding to smaller topology graphs that describe a small set of simple parametric roof shapes. Geometry for these simpler roof shapes can be described by a small number of parameters. The LIDAR points corresponding to the simple parametric shapes are used to estimate their geometry. Assuming that most buildings consist of rectilinear shapes, parts of a complex roof that are not explained by simple shapes can be modeled by fitting rectilinear shapes to the corresponding LIDAR points. Finally, in the terrain modeling stage, the segmented ground LIDAR points are smoothed and the terrain surface is estimated by a TIN created using Delauney triangulation.

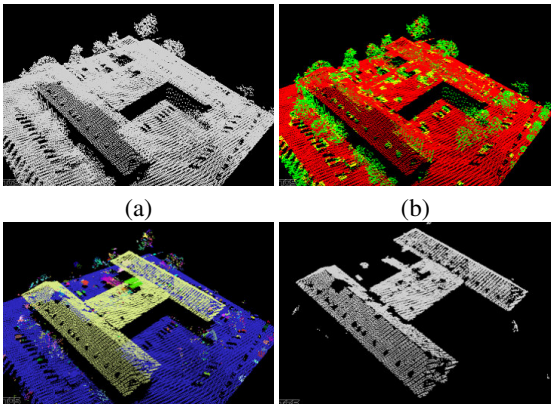
4. Roof and Terrain Segmentation

The LIDAR point cloud contains points that not only belong to the roofs of buildings but also other surfaces such as terrain, trees, cars, etc. In order to estimate the geometry of the buildings, we first need to identify the points that belong to the roofs. Note that the LIDAR data is not available for the sides of the buildings because the LIDAR instrument mostly sees only the roofs of the buildings. Also, the LIDAR point cloud density is expected to be fairly uniform over a given region.

Let $L = \{p_i | i \in [1, n]\}$ be n points in the LIDAR point cloud and $N_i = \{p_j | p_j \in L, distance(p_i, p_j) < \delta\}$ be the set of points in the δ-neighborhood of a point p_i (represented by a column vector). The 3×3 covariance matrix describing the 3D sub-space spanned by the points in N_i is given by

$$M_i = \frac{1}{|N_i|} \sum_{p \in N_i} (p - \mu_i)(p - \mu_i)^T$$

where, μ_i is the mean of all the points in N_i. If M_i is a matrix of rank 2 then the point p_i lies on a plane defined by the points in N_i. In practice, due to the presence of noise in LIDAR data and numerical round-off errors, the matrix M_i does not have rank 2 even when all the points in N_i lie on a plane. Rather than fitting a plane to the points in N_i, we simply compare the magnitude of the eigen values of M_i to determine whether the neighborhood of p_i is mostly flat. All points in the LIDAR point cloud L that are not locally flat are discarded. These non-flat points constitute most of the points that lie on non-flat surfaces such as trees, fences, poles, etc. Figure 2-(a) shows a LIDAR point cloud and Figure 2-(b) shows the result of identifying non-flat points. Since the LIDAR data does not strictly sample an urban area in the nadir direction, some points from the sides of vertical structures also end up in the LIDAR point cloud. The points on vertical surfaces are sparse because of the oblique angle between the LIDAR sensor and vertical building sides during an aerial LIDAR data collection. These points are identified by the dominant eigen vectors of their δ-neighborhood points. A point p_i that lies on a vertical wall would have the two dominant eigen vectors of matrix M_i such that they span a vertical plane. Points that lie on vertical surfaces are also removed from the point cloud. Let L_{flat} be the points from the original point cloud L that lie on locally flat and non-vertical surfaces. Once all the non-flat points have been removed from the point cloud, there is sufficient separation between points that lie on the roofs and points that lie on the ground. We next proceed to group all points based on their proximity to each other by finding connected components such that each point in a connected component is within a given distance to at least one more point in that component. Hence, all the points belonging to the roof of a building would lie in a single connected component. Since the ground is expected to contain the largest number of points, the largest connected component would contain all the points that belong to the ground. The LIDAR points do not lie at the corners of a fixed grid, hence we voxelize the volume containing the points in L_{flat} and use a 26-neighbor connected components method to identify points in different connected components. Voxel size is chosen to be v meters, assuming that the buildings are at least v tall and adjacent buildings are at least v apart. Small structures such as cars, low vegetation, mailboxes, etc. that are not very tall would also lie in the same connected component as the ground points. Figure 2-(c) shows the results of connected component analysis on the points in L_{flat}. Note that points are colored according to the connected component to which they belong. Also note that points belonging to some trees form a separate connected component. These are points that were not identified as non-flat because there



(a) (b) (c) (d)

Figure 2. (a) LIDAR point cloud. (b) Non-flat points are colored green, flat points are colored red. (c) Each connected component is colored differently. (d) Segmented roof points.

were enough points in their δ-neighborhood that formed a plane. These points can be eliminated by using an appropriate threshold because the number of points belonging to large roof structures are significantly greater then the number of points belonging to the trees. Figure 2-(d) shows the result of removing all components that have too few points as well as the largest component. After this step the original point cloud L is segmented into points that belong to the roof, L_{roof}, and points that belong to the ground, L_{ground}.

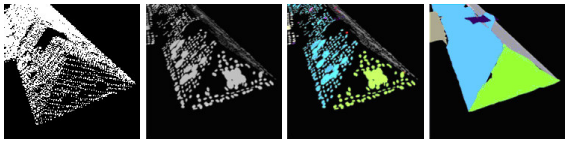
5. Building Detection and Recognition

5.1. Approximate Roof Modeling

Once all the points belonging to the roofs are identified, we build an approximate model of the roofs by identifying the individual planar roof surfaces as arbitrary polygons. This is achieved by fitting local planar patches to the identified roof points and grouping them based on the normals of these surface patches. A single plane is fit through each group of points and their boundary identified using the ball-pivoting algorithm [4]. Figure 3 shows the sequence of steps required to create an approximate roof model from segmented roof points L_{roof}. The polygonal roof segments thus computed are only an approximate representation of the actual roof surface because they do not form a watertight geometric model. Furthermore, the boundaries of these polygonal segments are very noisy resulting in very jagged building roof outlines. The next stage of our algorithm refines the approximate roof geometry computed in this section by recognizing and fitting simple parametric shapes to a complex roof shape.

5.2. Roof Topology Analysis

We now describe the algorithm to identify the different parametric shapes that can be combined to construct complex roof shapes. Given any roof composed of planar faces, we can describe the relationship between the various faces using a graph as described below.



(a) (b) (c) (d)

Figure 3. Approximate roof modeling. (a) Segmented LIDAR points for a roof surface. (b) Local plane patches fit to the LIDAR points. (c) Planar patches grouped together based on similarity of normals. (d) Approximate boundaries of planar roof patches.

- Each planar face of the roof is represented by a vertex
- Two vertices of the graph have an edge between them if the corresponding planar face are edge-adjacent (they share an edge)

The edges of the graph are labeled as follows depending on the normals of the polygons:

O^+ the polygons corresponding to the end vertices have normals that when projected on the XY plane are orthogonal and point away from each other.

O^- the polygons corresponding to the end vertices have normals that when projected on the XY plane are orthogonal and point towards each other.

S^+ the polygons corresponding to the end vertices have normals that when projected on the XY plane are parallel and point away from each other.

N no constraint

We call such a graph the *roof-topology graph*. Figure 4-(a) shows a simple roof shape (hip roof), and the corresponding roof-topology graph that describes the relationships between the four faces of the roof. Figure 4-(b) shows how the edge labels are determined for three of the above four cases. We will later show that labeling the edges of the roof-topology graph not only establishes geometric constraints between faces at the time of geometry fitting but also helps in recognizing sub-parts of a complex roof that can be modeled by simpler shapes (section 5.2.2). More complex roofs can similarly be analyzed to determine their topology graphs. Figure 5 shows a complex roof and the corresponding topology graph.

5.2.1 Roof-topology graph construction

The roof-topology graph of a roof shape can be constructed after the approximate boundaries of all the planar segments of the roof are estimated as described in Section 5.1. Note

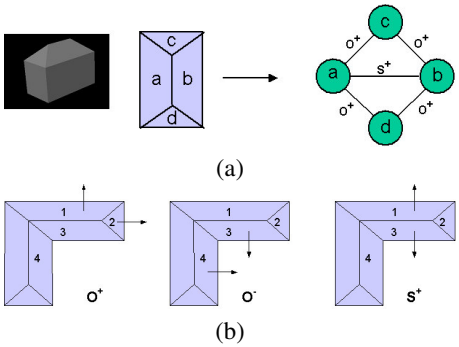


Figure 4. (a) A simple roof shape and the corresponding roof-topology graph. (b) Relationships between slopes of adjacent planar roof segments.

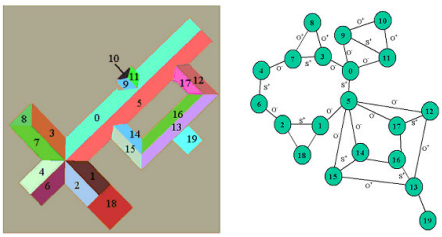


Figure 5. A complex roof shape and the corresponding roof-topology graph.

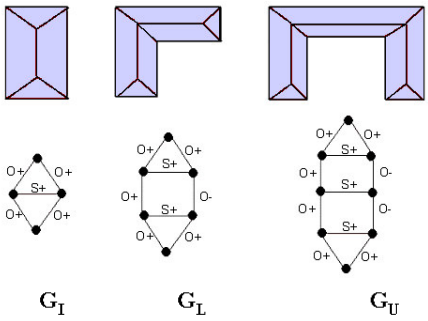


Figure 6. Roof-topology graphs corresponding to the simple parametric shapes that are used in the construction of complex roof shapes.

that the polygon boundaries in the approximate roof model are very jagged and polygons corresponding to two adjacent roof segments might not share an edge (Figure 7-(a)). A vertex is created in the roof-topology graph corresponding to each polygonal boundary. An edge is added between two vertices if the polygons corresponding to the two vertices are adjacent. In order to infer whether two planar segments are adjacent, we check whether any edge of the corresponding polygons are close to each other (Figure 7-(b)). For each edge in the roof-topology graph constructed thus far, we find the slopes of the polygons corresponding to the ver-

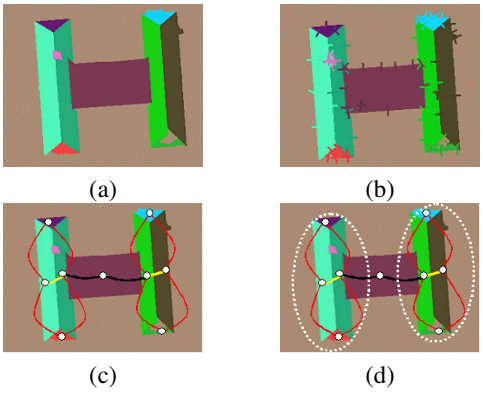


Figure 7. (a) Approximate roof shape. Note the jagged boundaries of the planar segments. (b) Searching in the vicinity of each polygon to to find its neighbor. (c) Roof-topology graph. Red edges represent O^+ labels, yellow edges represent S^+ labels, and black edges represent N labels. (d) The highlighted sub-graphs represent recognized parametric shapes.

tices and label the edge according to the rules described in section 5.2. Figure 7-(c) shows the resulting roof-topology graph.

5.2.2 Decomposing complex roofs into simpler shapes

As described in section 3, our approach relies on the observation that a complex roof shape can be constructed as a combination of several simple roof shapes. For example, the complex roof shown in Figure 1-(b) can be decomposed into multiple simpler roofs as shown in Figure 5-(right). In the previous section we introduced the concept of the roof-topology graph of a complex roof structure. It is important to note that the simple roof shapes can also be described by a roof-topology graph. The topology graphs corresponding to the simple roof types in Figure 1-(a) are shown in Figure 6. In order to recognize the simple shapes needed to model a complex roof, it is sufficient to recognize the roof-topology graphs corresponding to simple roof shapes as sub-graphs in the larger and more complex roof-topology graph corresponding to the complex roof shape, i.e. by sub-graph matching. While sub-graph matching is a NP-complete problem, we expect the graphs to be fairly small, hence a brute-force search to do sub-graph matching is quite fast and sufficient for our purpose. Let the sub-graphs corresponding to the shapes in Figure 6 be G_I, G_L, and G_U (G_L, G_U correspond to the L and U shape buildings, respectively) . To avoid any ambiguities in sub-graph matching, we recognize sub-graphs in decreasing order of complexity. Hence, we first find all sub-graphs that match G_U, followed by all sub-graphs that match G_L and G_I. This greedy approach ensures that we will decompose the complex roof-topology graph into as many sub-graphs as can be explained by G_U, G_L, and G_I. We next label the LI-

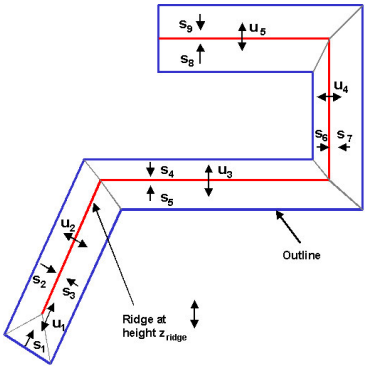


Figure 8. Ridge roof model free parameters

DAR points from L_{roof} as belonging to one of the simple parametric shapes. Remaining LIDAR points (parts of the roof-topology graph that are not explained by one of G_U, G_L, and G_I) are modeled as rectilinear shaped buildings.

6. Building Geometry Fitting

The foregoing topology extraction step collects the roof faces and associated LIDAR points into architecturally typical groupings (simpler parametric shapes). The roof refinement step described here fits a watertight polyhedral model to the LIDAR points. Since a global model is enforced, the result is likely to be more accurate and aesthetically pleasing than if each roof plane were independently estimated. The general procedure will be illustrated with the example of a ridge roof model that is a generalization of the three parametric shapes shown in Figure 6.

As depicted in Fig. 8, a ridge roof has planar faces, a ridge of N line segments at uniform height z_{ridge}, and an outline contour paralleling the ridge segments at uniform height z_{outline}. The shape of the outline is entirely determined by slicing the roof planes by a plane at height z_{outline}. Pairs of faces sharing a ridge segment have oppositely oriented gradients but not necessarily the same slope. Adjacent ridge segments do not necessarily meet at right angles. The ridge may end in either a sloped triangle face or a gable, *i.e.* a triangle–topped vertical wall section where the ridge meets the outline. Note that the ridge roof model subsumes the parametric shapes in Figure 6.

The roof fitting procedure consists of independently estimating each plane, hypothesizing symmetries, initializing a parametric model, and finally refining the parameters.

6.1. Estimating each plane

The parameters of each face are estimated from its associated LIDAR points by RANSAC, and outlier points are removed from further consideration. In case the roof ends in a gable, there are no sloped triangles to slice by z_{outline}. Thus, in order to define the outline segment at a gable end, the end segment is assumed to be perpendicular to the adjacent outline segments, passing through the outermost LIDAR point.

6.2. Hypothesizing symmetries

To improve the appearance and accuracy of the model despite noisy data, certain regularities are hypothesized and then enforced in the parametric model.

Pairs of faces sharing a ridge segment are forced to have opposing gradient orientations. Furthermore, if their independently estimated slopes are found to be close enough, then their slopes are forced to match. A consequence of matching slopes is that the ridge line will become centered between its corresponding parallel outline segments.

Frequently, parts of a building will be laid out at right angles to each other, *i.e.* many or all ridge segments would have a common orientation $\theta^* \pmod{90^\circ}$. To exploit this symmetry, θ^* is estimated from the mode of the distribution of face gradient orientations. Then, any face gradient that is close enough to $\theta^* \pmod{90^\circ}$ is forced to the common orientation.

6.3. Initializing parametric model

We shall fix the ridge segment and face orientations estimated above, as well outline height z_{outline}. The ridge roof model therefore has up to $3N + 5$ d.o.f., namely 1 for z_{ridge}, $2N + 2$ for face slopes $s_1 \ldots s_{2N+2}$, N for ridge line locations $u_2 \ldots u_{N+1}$, and 2 for ridge endpoint locations u_1, u_{N+2}. This count is reduced by 2 d.o.f. for each gable end, and by 1 d.o.f. for each pair of faces forced to have matching slopes.

Sets of three faces can be intersected to estimate 3D ridge vertex points. Their x, y values are used to initialize ridge location parameters u_j, and their median z value is used to initialize z_{ridge}. The slopes s_i are initialized from the independent plane estimates (possibly modified by slope matching constraints).

The subset of LIDAR points that form the rough outline of the roof is found by applying the 2D ball–pivoting algorithm[4] to the (x, y) component of all the inlier points. The median z value of that subset is taken to be z_{outline}.

6.4. Refining model parameters

The parameters are refined to minimize square error between the roof model and LIDAR points. The energy function to be optimized is a sum over all inlier LIDAR points (x_k, y_k, z_k):

$$E = \sum_k e_k^2 = \sum_k \left(s_i(a_i x_k + b_i y_k + u_j) + z_{\text{ridge}} - z_k\right)^2,$$

where s_i, u_j are the slope and ridge location for the face that LIDAR point k belongs to, and a_i, b_i are that face's fixed

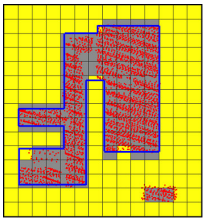


Figure 9. Finding the rectilinear outline of a set of points. The points are marked as red dots. A cartesian grid is overlaid on top of the points. Cells of the grid with points are colored gray. The refined rectilinear outline is shown in blue.

gradient orientation. The minimum energy with respect to the free parameters is found by Gauss–Newton optimization.

6.5. Fitting Rectilinear Shapes

In section 5.2.2 we described a method to explain as much of a complex roof as possible using the three variations of the ridge roof model estimated in the previous section. The remaining parts of the roof also need to be modeled using appropriate geometric primitives. Since most buildings have a rectilinear outline, it is reasonable to model remaining parts of the complex roof using rectilinear buildings. We also assume that each edge of the rectilinear building will be at least ξ units long. We choose the value of ξ to be 2 meters. The orientation of the rectilinear parts of a building is fixed to be the dominant orientation θ^* found in section 6.2. We first enclose the LIDAR points corresponding to the yet unexplained parts of the roof with an oriented bounding box (oriented to be aligned to θ^*). We next overlay a 2D grid over the LIDAR points in the XY plane, with spacing ξ, aligned with the oriented bounding box, as shown in Figure 9. The cells of the 2D grid are marked if it has any LIDAR points. Grid cells with very few LIDAR points are not marked. The boundary of the marked cells is an approximation of the rectilinear shape to model the unexplained parts of the roof. We first find the outline of the marked cell and then refine the boundary (move it parallel to itself) for a tighter fit around the LIDAR points in the marked cells. This process is illustrated in Figure 9. The roof slope and height of the rectilinear structure is found by fitting a plane to the LIDAR points and intersecting it with the rectilinear boundary.

6.6. Terrain Geometry Modeling

Terrain is modeled by fitting a triangle mesh to the terrain points L_{ground} as segmented from the original LIDAR point cloud (see section 4). The LIDAR point cloud subset L_{ground} does not contain any points from the buildings but it still contains points from other clutter elements such as cars as well as some points from low lying vegetation.

Since the exposed ground is typically smooth, we classify a 2D location (x, y) as exposed ground if the z values of LIDAR points in a 2D neighborhood of (x, y) span less than a set threshold. The mask of exposed ground points is eroded to remove noise, and then the LIDAR points lying in the eroded mask are used to interpolate a smooth uniformly gridded terrain. Specifically, the LIDAR points are sorted into 2D bins and the z values within each bin are averaged. Empty bins are filled by membrane interpolation, *i.e.* solving Laplace's equation $\nabla^2 z = 0$ in empty bins, with non-empty bins as boundary conditions.

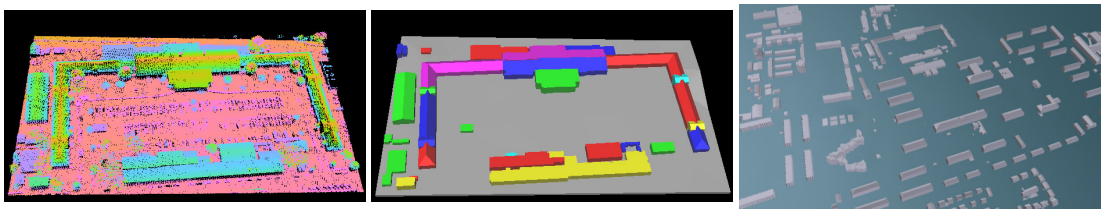
The urban area of interest, minus holes for the modeled building footprints, defines the 2D domain over which a terrain model is required. Given the piecewise linear boundary of that domain, we use Delauney triangulation to partition the domain into a planar mesh of triangles whose areas and vertex angles satisfy desired bounds. z values are assigned to mesh vertices from the foregoing uniformly gridded terrain.

7. Experimental Results

We have tested our algorithm on many different LIDAR sets taken over cities, small towns, suburban and rural areas. We report on experiments performed on a LIDAR data set collected at about 33 cms resolution over an urban area (9 samples/sq. m). The LIDAR data contained samples from 248 buildings. Our algorithm identified all the buildings and modeled them as a combination of 522 simpler shapes (including rectilinear shapes). We are able to build the complete model in about 10 minutes on a low-end PC with no user interaction at all.

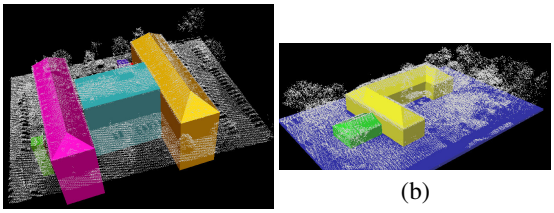
Figures 11 and 10 show results of automatic building and terrain modeling using our graph-based building recognition approach outlined in previous sections. Our segmentation algorithm is robust to the presence of clutter due to trees as shown in Figure 11-(b). Note that the area around the building in Figure 11-(b) has a lot of trees. Our algorithm successfully recognized and modeled the building even when a large portion of the roof was covered by parts of trees adjacent to the building. While our roof topology inferring algorithm correctly recognized most of the simple sub-shapes of complex roof shapes, it sometimes failed to correctly recognize parts of some buildings as one of the parametric shapes. This happened due to insufficient LIDAR samples on part of a roof segment, trees obscuring large parts of a roof, or non-uniform sampling of parts of a roof surface. Some of these errors stem from the limitations of LIDAR data itself – non-uniform sampling, insufficient resolution around surface edges, poor reflectance from some surfaces, etc.

The simple parametric shapes used in our approach can model a large number of typical building roofs but they also fail to model some class of simple shapes such as pyramids, domes, etc. This problem can be rectified by expanding the set of parametric shapes used to fit sub-shapes of complex roofs. For example, the roof-topology graph of a pyramid shape roof is a graph that consists of a cycle with four ver-



(a) (b) (c)

Figure 10. (a) Example of a LIDAR point cloud where each point is colored based on its height. (b) Result of automatic building geometry modeling for the LIDAR point cloud in (a). Each sub-part of the building recognized as a simple shape is colored differently. (c) Buildings modeled for a large urban area.



(a) (b)

Figure 11. Results of recognizing complex buildings as a combination of several simple shapes. Each simple shape is colored differently.

tices and all edges labeled as O^+. Our framework also allows a user to incrementally add to the set of simple parametric shapes by creating the roof-topology graphs corresponding to new shapes. Hence our method is not limited by the set of parametric shapes chosen in advance.

8. Conclusions

This paper presents an algorithm for rapid and automatic 3D building detection and geometric modeling using aerial LIDAR point cloud data. The LIDAR points belonging to building roofs and terrain surfaces are separated from clutter due to trees etc. by locally analyzing the neighborhood for each point. Planar regions are grown using a 3D connected components analysis to identify continuous smooth regions. A novel contribution of this work is the formulation of a powerful framework to analyze topology of complex roof shapes using the roof-topology graph. The problem of recognizing simple parametric shapes that can be combined to construct a given complex roof is reduced to the problem of sub-graph matching. We have demonstrated that a small vocabulary of simple parametric shapes can be used to express very complex roof structures. Future improvements include the adaptation of the current framework to include non-flat roof shapes such as domes, automatically infer constraints between different sub-parts of a building, and refine geometry for the sides of buildings using ground-based LIDAR.

Acknowledgement: We thank ONR and PMTRASYS for supporting this research (contract number NMA201-01-C-0003).

References

[1] S. Ahlberg, U. Soderman, M. Elmqvist, and A. Persson. On modeling and visualization of high resolution virtual environments using lidar data. In *Proceedings of the 12th Int. Conf. on Geoinformatics*, pages 299–306, 2004.

[2] A. Alharthy and J. Bethel. Heuristic filtering and 3d feature extraction from lidar data. In *PCV02*, page A:29, 2002.

[3] C. Baillard, C. Schmid, A. Zisserman, and A. Fitzgibbon. Automatic line matching and 3d reconstruction of buildings from multiple views. In *ISPRSGIS99*, pages 69–80, 1999.

[4] E. S. de Medeiros Filho, L. Velhoa, and H. Lopes. Restricted BPA: Applying ball-pivoting on the plane. In *XVII Brazilian Symposium on Computer Graphics and Image Processing*, pages 372–379, 2003.

[5] P. E. Debevec, C. J. Taylor, and J. Malik. Modeling and rendering architecture from photographs: A hybrid geometry- and image-based approach. In *SIGGRAPH*, pages 11–20, 1996.

[6] A. Elaksher and J. Bethel. Reconstructing 3d buildings from lidar data. In *PCV02*, page A: 102, 2002.

[7] C. Früh and A. Zakhor. Constructing 3d city models by merging ground-based and airborne views. In *CVPR 2003*, pages 562–569, 2003.

[8] N. Haala and C. Brenner. Extraction of buildings and trees in urban environments. *PandRS*, 54(2-3):130–137, July 1999.

[9] J. Hu, S. You, and U. Neumann. Approaches to large-scale urban modeling. *IEEE CG&A*, 23(6):62–69, 2003.

[10] S. C. Lee and R. Nevatia. Interactive 3d building modeling using a hierarchical representation. In *HLK*, pages 58–65, 2003.

[11] G. Vosselman. Building reconstruction using planar faces in very high density height data. In *International Archives of Photogrammetry and Remote Sensing*, pages 87–92., 1999.

[12] S. You, J. Hu, U. Neumann, and P. Fox. Urban site modeling from lidar. In *ICCSA (3)*, pages 579–588, 2003.