

# ROBUST EXTRACTION OF EXTERIOR BUILDING BOUNDARIES FROM TOPOGRAPHIC LIDAR DATA

Stephen R. Lach<sup>a,b</sup> and John P. Kerekes<sup>b</sup>

<sup>a</sup> United States Air Force

<sup>b</sup> Digital Imaging and Remote Sensing Laboratory, Rochester Institute of Technology  
54 Lomb Memorial Drive, Rochester, NY 14623  
[sf11194@cis.rit.edu](mailto:sf11194@cis.rit.edu), [kerekes@cis.rit.edu](mailto:kerekes@cis.rit.edu)

*DISCLAIMER: THE VIEWS EXPRESSED IN THIS ARTICLE ARE THOSE OF THE AUTHORS AND DO NOT REFLECT THE OFFICIAL POLICY OR POSITION OF THE U.S. AIR FORCE, DEPARTMENT OF DEFENSE, OR U.S. GOVERNMENT.*

## ABSTRACT

Methods for generating accurate building models from lidar data have received considerable attention in the recent literature. Many of the proposed techniques examine the data to define dominant planes in the structure, then intersect these planes to determine the location of internal edges and vertices. However, since most airborne lidar datasets are collected from near-nadir orientations, there are usually very few data points that lie on vertical surfaces. As such, it is usually difficult to determine the planes corresponding to exterior walls using data points on these walls. However, if we assume that exterior walls are oriented directly under the outer boundary of the roof structure, we may identify the geometry of these walls by modeling the 2D shape of the building exterior. This paper presents a robust approach for extracting this exterior boundary directly from the lidar data.

*Index Terms*— lidar, shape, buildings, modeling

## 1. INTRODUCTION

Accurate 3D modeling of the environment from remotely-sensed data has found applications in such diverse fields as urban planning, homeland security, environmental research, computer graphics and noise propagation. One of the more difficult tasks in creating these scene models has been automating the recovery of building geometries. With the recent advances in lidar sensing technology, autonomous and semi-automated building modeling from active imagery has received much attention in recent years. A typical approach is to first isolate building points from their surroundings, then to identify dominant planes in this data. These planes are then intersected to determine internal edges and vertices. However, due to the viewing geometry of most airborne lidar data collections, very little data is

available on vertical wall surfaces, so a different technique must be used to determine the building's external boundary. This paper describes our approach to first identifying the external boundary points of a potentially non-convex bounding polygon, followed by a simplification/regularization algorithm that robustly adjusts this data according to pre-defined constraints.

## 2. BACKGROUND

There are several methods of extracting and simplifying building boundaries from lidar data in the literature. For example, in [1] the authors use data that has been interpolated to a grid, then extract the boundary through morphological techniques. A point-merging algorithm is then used, where points are successively deleted if their removal does not change the final geometry of the boundary by more than a given threshold. In [2], the author also uses morphology to extract building boundaries from a rasterized dataset. However, in this approach, a constraint is placed on edges such that they are forced to be parallel or perpendicular to the dominant building orientation. The authors of [3] work with raw lidar points, and the hierarchical least-squares regularization they propose relaxes the constraint that all building edges must align with dominant edges. However, in this case, edges that are out of alignment may influence the orientation of other edges.

## 3. APPROACH

Like the technique proposed in [3], our technique works on raw point data, thereby removing the errors inherently introduced in the interpolation process. However, a novel method is introduced to improve the orthogonality constraint, while preventing edges that are not orthogonal or parallel to the dominant building orientation from

influencing the adjusted orientation of the *conforming* edges.

In this approach, once data representing a single building are isolated from the surrounding points, the potentially non-convex set of exterior points are identified and ordered. This set is then reduced through a *line simplification* process, where points not critical to the definition of the final shape are removed. Finally, a *boundary regularization* step is performed, where the dominant orientation of the building is determined, and certain edges are realigned to fall either parallel or perpendicular to the dominant orientation.

### 3.1. Determining the Outer-most Data Points

For the purposes of this paper, it will be assumed that the lidar data pertaining to a particular building of interest have already been isolated from the surrounding points. As such, the first task in the boundary extraction process is to isolate the exterior data points that effectively describe the intuitive 'shape' of the building.

In order to accomplish this, many researchers make the assumption that buildings have a strictly convex boundary geometry. An example of such an approach is given in [4], where the authors use the exterior triangles from a Delaunay facetization to determine outer boundary line segments. However, such techniques fail for more complex geometries. In Fig. 1, we present a set of points representing the outer boundary of an 'L-shaped' building projected into the ground plane. The connectivity shown in (a) represents the result of using a Delaunay-based convex hull approach. This method fails to highlight all of the boundary points, and more importantly, it misses the interior angle.

In order to deal with this shortcoming, an alternate definition for the bounding shape of a point set must be used. To this end, we have opted to use  $\alpha$ -shapes for the determination of exterior object boundaries. Like the convex hull,  $\alpha$ -shapes are simply an approach to formally describe the 'shape' of a set of spatial point data. However, unlike the convex hull,  $\alpha$ -shapes are not limited to convex geometries,

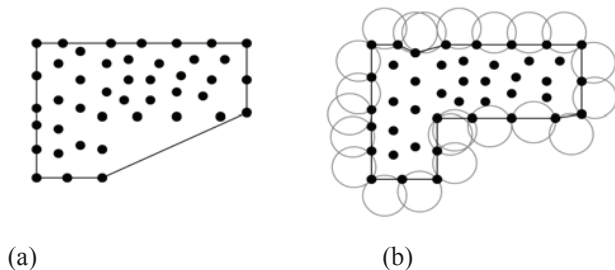


Fig.1 Defining the bounding shape of a set of points: (a) Convex hull and (b)  $\alpha$ -shapes.

and may even represent holes inside the geometry. Additionally, for a given point set, the resultant  $\alpha$ -shape is a function of a pre-determined structuring element termed the  $\alpha$ -ball. Therefore, the final 'shape' is actually an entire family of shapes.

Per [5], alpha-shapes may be computed directly from a 2D Delaunay triangulation of the data. This is done by removing all edges and triangles that have circumscribing spheres with radius greater than  $\alpha$ . The  $\alpha$ -complex is the portion of the Delaunay triangulation that remains, and the  $\alpha$ -shape is the boundary of the  $\alpha$ -complex [6]. Figure 1 (b) depicts the the  $\alpha$ -shape for the set of 2D data points given the  $\alpha$ -ball shown in gray.

### 3.2. Line Simplification

Once the  $\alpha$ -shape points are extracted, we have an initial estimate of the building boundary. However, since this shape typically has an irregular geometry, it is usually undesirable to use this as the final building boundary. In general, a line simplification algorithm must be employed to produce less noisy results

To this end, we implemented an algorithm based on the sleeve-fitting approach described in [7]. Traditionally the approach has been regarded as slightly less effective than the Douglas-Peucker algorithm [8], but for the task at hand it performs quite well. This algorithm was first applied to the building boundary problem in [2], but both Ma's work and the present research make significant modifications to the baseline approach described in [7].

Although a recursive version of the algorithm is presented in [7], the geometry of the basic sleeve algorithm is best understood through a brief description of the process. The width of the sleeve is selected at the outset, and this determines the maximum deviations that will be permitted before new line segments are defined. Starting at a chosen point  $P_1$  (which is also labeled as a critical point), we then consider the distance of point  $P_2$  from the line segment connecting  $P_1$  and  $P_3$ . If this distance is less than the sleeve width,  $P_2$  lies within the sleeve and is considered a non-critical point. In the next iteration, we consider the segment  $P_1$  to  $P_4$ , as shown in Fig. 2 (a). If all the intermediate points fall within the sleeve (which is the case in the figure),  $P_3$  is deemed non-critical.

Moving to the next iteration, we connect  $P_1$  to  $P_5$  and see that intermediate points now fall outside the sleeve, as in Fig. 2 (b). As such, the previous point,  $P_4$  is declared a critical point. The process begins again starting at the location of this new critical point. Therefore, we now connect  $P_4$  to  $P_6$ , and find that  $P_5$  lies outside the sleeve. As such,  $P_5$  is also labeled a critical point. The method continues with  $P_6$  being connected to  $P_8$ , and since  $P_7$  lies within the sleeve, it is determined to be non-critical. This process is then continued until all points have been examined.

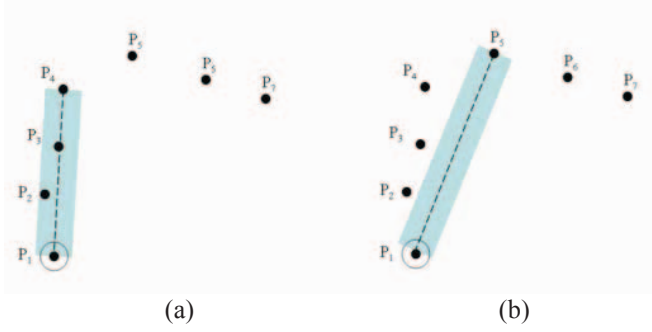


Fig.2 Illustration of the Sleeve Method for line simplification

We have modified the basic sleeve approach somewhat in order to select better critical points. Our first improvement is that we run the algorithm multiple times. First, the algorithm is run to find the initial set of critical points. We then run the algorithm again, starting at the third critical point, in order to get a better characterization of the original starting location. Finally, we perform two additional runs in the reverse direction to compensate for a potential angular bias that can occur with certain building geometries.

Also, in the original reference, Zhao uses the segments connecting the critical points as the simplified boundary, discarding the points in between. However, similar to [2], we improve the accuracy of the simplified lines by using an orthogonal regression through the points between (and including) the critical points. These regressions produce lines, and consecutive lines are then intersected to produce a new set of critical points. In general, these final critical points do not lie at locations covered by the original data set, but they will more accurately represent the location of building corners.

### 3.2. Boundary Regularization

In the vast majority of buildings, the outer boundary contains many edges that lie either parallel or perpendicular to the dominant building orientation. We may use this fact to our advantage in refining our simplified building edges through a process termed boundary regularization.

In the present work, we have created an algorithm that forces boundary line segments to be either parallel or perpendicular to the dominant building orientation when appropriate, and to fit the data without constraint elsewhere. Additionally, the uniquely oriented segments do not influence the calculation of the primary building orientation. A unique aspect of this new approach is that all conforming segments are initially brought into a common space where a segment and one perpendicular to it would have the same orientation. This allows the algorithm to work on all conforming segments simultaneously, and is not a weighted

average of the solution defined for each of two orientation classes.

To begin our approach, we first collect all of the line segments defined in the line simplification process (we term the set  $S$ ), and re-align them so that their azimuthal orientation angle is in the range from 0 to 90 degrees. The entire set of (potentially rotated) segments is termed  $S_r$ . Note that a record is kept for each vector as to whether or not it has been rotated, so later in the process this rotation may be undone.

The next step is to identify all line segments in  $S_r$  with a length  $\ell$  greater than a given threshold (typically 3m) to use for the initial processing. We term this set of segments  $S_{long}$ . If at least four segments of the required length are not available, all segments are assigned to  $S_{long}$ . We then step through each segment of  $S_{long}$  in turn and determine the set of  $S_{long}$  segments that are within  $15^\circ$  of current segment. The set of segments for the  $i^{th}$  segments is termed  $S_i$ , and the  $S_i$  with the largest cumulative length is selected and redefined as  $S_a$ .

A weighted average angle is then obtained from the segments of  $S_a$  according to

$$\theta_a = \frac{\sum_{i \in S_a} \ell_i \theta_i}{\sum_{i \in S_a} \ell_i} \quad (1)$$

where the  $\ell_i$  and  $\theta_i$  values are the lengths and azimuthal angles of the line segments in  $S_a$ . This is the dominant orientation of the set  $S_{long}$ . We now define the primary orthogonal azimuth as  $\theta_b = \theta_a \pm 90^\circ$ , using whichever sign keeps the orientation within the range  $-90^\circ \leq \theta_b < 90^\circ$ .

Next, we return to the initial (unrotated) set  $S$  and evaluate an updated azimuth for each segment. Thresholds are defined for both angular matching (typically  $15^\circ$ ) and for minimum length (typically 4m). Segments below the length threshold are assigned to the closest angular class (that is, they are given the slope  $\theta_a$  or  $\theta_b$  while keeping the location of the centroid point unchanged), regardless of the degree of angular similarity. This assignment is made since it is fairly common for the line simplification algorithm to detect multiple critical points at a single corner if the boundary data is severely contaminated with spatial noise. By forcing all small segments to align with the dominant building orientation, this effect is mitigated. Segments that are longer than the length threshold have their slope modified to  $\theta_a$  or  $\theta_b$  if their original slope is within the angular threshold of the potentially assigned slope value. Segments longer than the length threshold that have an angular difference greater than the angular threshold are left unchanged.

Finally, adjacent segments whose slopes are the same are combined into a single segment if the orthogonal distance between the segments is less than 3m. If the

orthogonal distance is greater than this threshold, a perpendicular connecting segment is added, and the user is flagged that there may be an issue with the boundary regularization process at that location. In order to combine adjacent segments having the same slope, in a manner to the one described in [2], the centroid of the combined segment is the weighted average of the centroids of the two segments, where the segment lengths are used as weight values and the slope is kept unchanged.

#### 4. RESULTS

Fig. 3 presents an illustration of the boundary extraction process as applied to real data from a typical suburban residence. To illustrate the full technique, this data has undergone one modification; a triangular section of points in the upper-right corner have been removed so as to create a uniquely oriented edge. In (a) we see the original point data projected to the ground plane, and (b) shows the points selected by the  $\alpha$ -shape extraction. The sleeve algorithm is then run a total of four times (two forward and two reverse) and the critical points shown in (c) are identified. In (d) we see the outcome of the robust regularization algorithm. This final illustration shows that we are achieving the desired result. The uniquely oriented edge has been modeled appropriately, and all smaller edge segments have been adequately merged with longer edges. The conforming edges are seen to remain mutually perpendicular, and they fit the data without any angular bias induced by the uniquely oriented edge. Additionally, all corner points appear at their correct locations, even if the original dataset did not contain points at these positions.

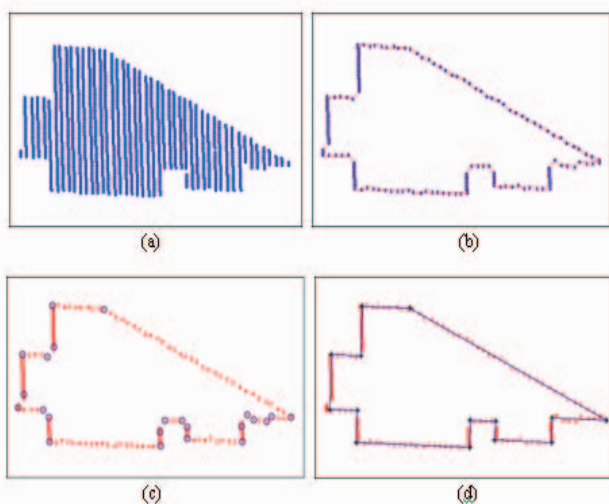


Fig. 3. Example of the boundary extraction process

#### 5. SUMMARY

This work presented an overview of our method for autonomously extracting the exterior boundary of a building from lidar point data. Through an application of  $\alpha$ -shapes, an initial set of potentially non-convex external building points was identified. This bounding set was then filtered using the modified sleeve technique, and a set of critical points representing transition regions was defined. Finally, a robust regularization algorithm was used to ensure that conforming edges were constrained to be mutually perpendicular while not being biased by uniquely oriented edges.

In the research presented here, we aimed to increase the accuracy of automated building modeling from lidar data without fusing the data with other sources of information. Future work in this area will seek to improve the accuracy of the defined building boundaries through the incorporation of additional imaging modalities. Additional detail regarding the techniques discussed here, as well as expanded results, may be found in [9].

#### 6. REFERENCES

- [1] Weidner, U. and W. Forstner. "Towards automatic building reconstruction from high resolution digital elevation models." *ISPRS Journal of Photogrammetry & Remote Sensing* 50, pp. 38-49, 1995.
- [2] Ma, R., *Building Model Reconstruction from Lidar Data and Aerial Photographs*, PhD thesis, The Ohio State University, 2004.
- [3] Sampath, A. and J. Shan. "Building Boundary Tracing and Regularization from Airborne Lidar Point Clouds." *PE&RS* 73, pp. 805-812, July 2007.
- [4] Chen, Teo, Rau, Liu, and Hsu. "Building Reconstruction from LIDAR Data and Aerial Imagery." *Proc of IEEE, IGRSS '05*, Vol 4, pp. 2846-2849, 2005.
- [5] Edelsbrunner, H., D. Kirkpatrick, and R. Seidel. "On the shapes of a set of points in the plane." *IEEE Transactions on Information Theory* IT29, April 1983.
- [6] Kavraki, L. Jun. 2007 "Molecular Shapes and Surfaces." Web-tutorial: <http://cnx.org/content/m11616/latest/>, 2008
- [7] Zhao, Z. and A. Saalfeld. "Linear-Time Sleeve-Fitting Polyline Simplification Algorithms." *AutoCarto 13*, Seattle, Washington, pp. 214-223, 1997.
- [8] Douglas, D. and T. Peucker. "Algorithms for the reduction of the number of points required for representing a digitized line or its caricature." *Canadian Cartographer* 10, pp. 112-122, 1973.
- [9] Lach, S., *Semi-Automated DIRSIG Scene Modeling from 3D Lidar and Passive Imagery*, PhD thesis, RIT, 2008.