# Segmentation and Reconstruction of Polyhedral Building Roofs From Aerial Lidar Point Clouds

Aparajithan Sampath and Jie Shan, *Member, IEEE*

***Abstract*—This paper presents a solution framework for the segmentation and reconstruction of polyhedral building roofs from aerial LIght Detection And Ranging (lidar) point clouds. The eigenanalysis is first carried out for each roof point of a building within its Voronoi neighborhood. Such analysis not only yields the surface normal for each lidar point but also separates the lidar points into planar and nonplanar ones. In the second step, the surface normals of all planar points are clustered with the fuzzy $k$-means method. To optimize this clustering process, a potential-based approach is used to estimate the number of clusters, while considering both geometry and topology for the cluster similarity. The final step of segmentation separates the parallel and coplanar segments based on their distances and connectivity, respectively. Building reconstruction starts with forming an adjacency matrix that represents the connectivity of the segmented planar segments. A roof interior vertex is determined by intersecting all planar segments that meet at one point, whereas constraints in the form of vertical walls or boundary are applied to determine the vertices on the building outline. Finally, an extended boundary regularization approach is developed based on multiple parallel and perpendicular line pairs to achieve topologically consistent and geometrically correct building models. This paper describes the detail principles and implementation steps for the aforementioned solution framework. Results of a number of buildings with diverse roof complexities are presented and evaluated.**

***Index Terms*—Building extraction, clustering, LIght Detection And Ranging (lidar), reconstruction, segmentation.**

## I. Introduction

BUILDINGS are an indispensible component for a 3-D geospatial information system. LIght Detection And Ranging (lidar) technology provides dense accurate georeferenced 3-D point clouds over reflective objects, from which building models can be derived. Although available auxiliary data such as building plan maps can assist this process [1], they may not always be available or up to date. Therefore, using lidar data alone have attracted most of the efforts in building extraction. Among the reported studies, polyhedral building is a quite common assumption, i.e., the building has only planar roofs, based on which various model detection methods in pattern recognition can be adopted for building extraction.

Methodologically, building extraction involves a more fundamental and broader concept called segmentation. It refers to the task of dividing a given data set into nonoverlap homogenous regions that constitute the entire data set [2]. Classical segmentation process is data driven and falls into two categories, both originated and primarily used in image processing and later adapted to range data processing. The first employs the region growing technique, while the other delineates regions by detecting edges in the data set. Region growing approaches start with a selected seed point, calculate its properties, and compare them with adjacent points based on certain connectivity measure to form the region. Edge-based methods determine the edges in the data set and connect them to form regions. These edges may be step edges that represent a jump in the range data or ridges at two different surfaces. Segmentation is attempted by closing the detected edges.

Both edge-based and region growing methods have been used for range data segmentation, with the latter being more popular. Reference [3] determines edges using local surface curvature properties to detect zero crossings. The region defined by edges is then refined by using quadric patches. Reference [4] uses a local convex hull approach to detect edges to segment the building boundary and roof planes. A common shortcoming of the edge-based methods is that it is susceptible to outliers and incomplete edges that do not form explicit segments. As an early study on region growing methods for range data segmentation, [5] divides a scan line into line segments and use adjacent segments, three at a time, as a seed for the subsequent region growing algorithm. Recently, region growing methods are applied to building extraction [6]–[10]. As a typical implementation of region growing methods, [11] detects planes in small neighborhoods in the point clouds and use a connected component analysis to extend these planes. It makes use of the concept of roof topology, including adjacency and connectivity, to constrain a building to a combination of basic building models. This approach is certainly more efficient than fitting models directly to the data, as it is easier to detect models from planes other than directly from points. However, several constraints have to be applied to account for the structural variety of buildings, such as overhangs and roof structures on top of each other. Reference [12] combines surface modeling with an iterative procedure for seed point identification. The seed points are initially selected by using a local planar fit on the points. The region is expanded by fitting points that give a smaller residual error than the noise determined for such a fit. A combination of edge and region growing algorithm attempts to take advantage of the strengths of both methods, while avoiding the pitfalls. Reference [13] divides the data set into 3-D grids

Manuscript received December 21, 2008; revised May 29, 2009. First published November 3, 2009; current version published February 24, 2010.

A. Sampath is with the School of Civil Engineering, Purdue University, West Lafayette, IN 47907 USA, and also with the Stinger Ghaffarian Technologies, Inc., U.S. Geological Survey's Center for Earth Resources Observation Science, Sioux Falls, SD 47918 USA (e-mail: asampath@usgs.gov).

J. Shan is with the School of Civil Engineering, Purdue University, West Lafayette, IN 47907 USA (e-mail: jshan@purdue.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/TGRS.2009.2030180

0196-2892/$26.00 © 2009 IEEE

in an octree representation. The grids are created based on the surface normal at each point. As the grid size becomes smaller, the indication is that these grids are located near edges, and are removed from the octree. The region growing technique then uses the adjacency of the grids to coalesce based on the surface normals. Reference [14] presents an approach to segmenting and modeling point cloud generated from computer-aided design models. In this case, instead of an octree, a mesh is used for adjacent points. Edges detected using curvature are removed, and the remaining mesh elements are grouped together by using a region growing algorithm such that the regions do not include the border mesh elements. Despite different versions of implementations, a common complaint against region growing is that it usually fails when transitions between two regions are smooth and the parameters needed to stop the region growing are difficult to determine. In addition, problematic is the process of identifying noise-free seed regions.

Instead of data-driven segmentation methods, model-driven segmentation methods are often used when the mathematical expression of the objects to be extracted is known. This is closely related to the model detection theory in pattern recognition and machine learning. One of the popular model recognition methods is the Hough transform [15]. It can be used to determine roof plane parameters [16]. The common practice is to define a plane using three parameters $\{\theta, \varphi, \rho\}$, where $\rho$ is the distance of the plane to the coordinate system origin, and $\cos\theta, \cos\varphi$ are the directional cosines of the plane. The process consists of discretizing the values of $\theta$, $\varphi$ between 0° and 90°, and 0° and 360°, respectively, and then evaluating the values of $\rho$ for the point clouds. The triplet $\{\theta, \varphi, \rho\}$, each representing a unique plane, is analyzed for recurring counts. The triplets that reach certain number of counts are regarded as possible planes. However, this process results in spurious planes when processing lidar point clouds. Most of the implementations therefore use additional constraints. Reference [17] combines the process with building ground plans to eliminate some spurious planes and then project the remaining planes onto 2-D axial planes to further investigate and eliminate possible spurious planes. Reference [18] tests the ratio between the plane area and its number of points to discard spurious planes.

Another popular method for extracting models from a data set is the RANdom SAmple Consensus (RANSAC) approach [19]. It randomly and iteratively samples the least number of data points necessary to determine the model parameters. The derived parameters are then tested against the rest of the data set. The process stops when a sufficient number of data points from the rest of the data set fit with the parameters. At this time, the parameters are reestimated with the new set of data points. The RANSAC method needs to be applied to the entire data set and generates only one set of parameters when the process is finished [20]. Therefore, to extract polyhedral building roofs with multiple planes, the process has to be used iteratively and in a subtractive manner. This means that the points that have been shown to belong to a particular plane are excluded from the next RANSAC iteration. The RANSAC process has been used to estimate polyhedral building models, although it can lead to false planes, particularly when being used without any additional constraints and when the data set is noisy. Although it is an iterative process, [21] suggests it is faster than the Hough transform. It improves the RANSAC plane segmentation process by having a limit on the minimum number of points for a plane segment and the standard deviation in the final fitted plane. Reference [10] uses a combination of region growing process and RANSAC to determine roof planar segments. The data (in raster form) are used to estimate the slopes at all pixels and the pixels with similar slopes are grouped together. A RANSAC process is used to determine the points within each group that belong to a single plane. Determining the slopes in a noisy data set or near regions of ambiguity such as breaklines and step edges, vertical walls, trees, chimneys, etc., is a difficult process. The process of grouping pixels of similar slopes may not be simple because of the presence of these outliers, and the groups may turn out to have a larger than optimal “tail.” The RANSAC process may alleviate the issue, but since it is used to determine different planes from a single grouping, it can result in some spurious planes.

Classification or clustering techniques can also be used for segmentation of lidar points. Such techniques look for patterns in the data set. In case of building reconstruction, the patterns are homogenous surfaces, such as planes, separated by breaklines. A feature vector is defined to characterize the object to be extracted as uniquely as possible. In case of planar surfaces, the feature vector at each lidar point can consist of the surface normal and location of the point [22]. The normal vector can be generated by selecting a neighborhood around a selected location and fitting a plane based on the least squares criterion. References [23]–[26] demonstrate the use of clustering techniques for building extraction. Reference [23] uses a slope adaptive neighborhood to generate the feature vectors. Reference [24] clusters feature vectors by generating a tangential plane (normal vectors) and a height difference measure. The choice of neighborhood assumes importance for generating the tangential planes as they may be affected by the presence of noise and outliers. In [25] and [26], the data set is divided into patches, each of which is evaluated with respect to its planarity. The determined planar patches are then clustered into roof planar segments based on their normals and connectivity. Reference [27] generates the Gaussian sphere formed by the unit normal vectors of all data points and then determine the clusters of the normal vectors using a mean shift clustering process. Different types of clusters are identified on the Gaussian sphere, including pointlike, curvelike, and arealike clusters. Regions are then segmented based on priorities for the clusters, with pointlike (planar) clusters receiving the highest priority. The disadvantage of this technique is that the assumption on the number of segments is rather subjective, and the clustering process is susceptible to noise, which includes both sensor noise and model bias.

It should be noted that lidar data will invariably contain returns from regions that may not be planar, for instance from parts of trees, arches, chimneys, etc. This is in addition to the sensor and other kinds of noise present in the measurements. Such returns represent biases from the assumption of planarity and hence need to be removed from the segmentation process. The segmentation algorithms described above do not necessarily take into account the diversity of the spatial nature of the lidar data sets.

This paper presents a solution framework for building roof extraction. It tries to minimize the shortcomings of the aforementioned segmentation algorithms. We first apply eigenanalysis to every lidar point to determine its planarity. Nonplanar points, such as trees, roof edges, and arches are excluded from the subsequent clustering to warrant a robust solution. This avoids the general pitfalls of most region growing and clustering processes, while taking advantage of the edge-based approach. It should be noted this strategy is also consistent with the human vision process, in which objects are mostly perceived based on their edges and boundary [28]. In the next step, the fuzzy $k$-means approach is adopted to cluster the planar points into roof segments based on their surface normals. As a measure of cluster similarity, the distance criterion in this process considers both geometry (the Euclidean distance in feature space) and topology (the planarity of the points). This method also allows for a determination of the number of planar segments of a building by using a potential-based iterative clustering process. The final roof reconstruction explicitly determines the location and connectivity of roof boundary, ridges, and vertices based on the segmented planes.

In the remaining content of this paper, Section II discusses neighborhood selection, surface normal calculation, and breakline detection. Section III describes a potential-based fuzzy $k$-means clustering technique to determine the lidar points that belong to the same class. A density-based connectivity analysis allows for further separation of the coplanar segments. The properties and performance of the clustering process are discussed and evaluated in Section IV. In Section V, the topologic relations among the planar segments are represented by an adjacency matrix, which is used to determine roof ridges, edges, and vertices of the building model for final reconstruction. Results of segmentation and reconstruction for buildings with a variety of complexities are presented and evaluated. Section VI presents the final discussion and conclusion on the techniques presented in this paper.

## II. Surface Normal Vectors

One of the representations of a surface is the envelope of its tangential planes [29]. Since the surface normal vectors define these tangential planes, surface normal determination becomes one of the first steps in surface reconstruction. This section will address the neighborhood selection and dimensionality analysis for such purpose.

### A. Neighborhood Selection

Let $\mathbf{P}$ represent the subset of lidar points that belong to the roof of a single building. To estimate the surface normal at a point $\mathbf{p} \in \mathbf{P}$, a set of neighborhood points of $\mathbf{p}$ needs to be chosen. A large neighborhood may result in the loss of the local characteristic of the surface, while a small one will lead to an insufficient number of points for reliable surface determination. Therefore, it is critical to select a proper neighborhood. Reference [30] discusses the normal vector estimation for a smooth surface from discretely sampled points, and suggests using the Voronoi neighborhood $V_p$ of the point $\mathbf{p}$. It is justified that the Voronoi meshing creates a unique set of neighbors for each point and the Voronoi meshing process does not depend on the point density. An example of such Voronoi neighborhood for a lidar point is shown in Fig. 1, where the Voronoi neighborhood of the circled point is shown by the squared dots. This paper will use the Voronoi neighborhood $V_p$ for surface normal calculation.

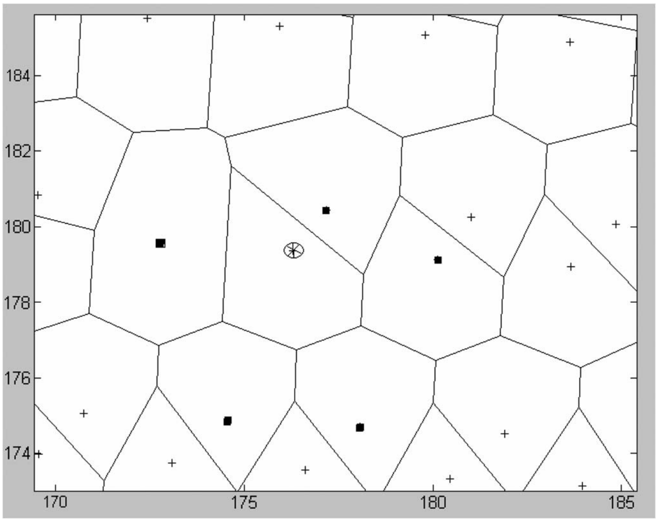


Fig. 1. Voronoi neighbors (square dots) of the circled dot. The Voronoi cells are shown in lines and the cross $(+)$ dots are other lidar points in the point cloud. The two axes are in meters.

It should be noted that the characteristics of the aerial lidar data and the object of this paper (building roof) allow for using 2-D Voronoi polygons to define the neighborhoods. Except regions near the vertical walls of a building, the point cloud can be approximated by a 2-D surface. There are, if any, only a few lidar returns from the vertical sections of a building, which are usually insignificant comparing to the returns from the building roof. As will be discussed in the next section, normal vectors for such points, i.e., points that lie in the neighborhood of more than two planes, will be discarded for robust clustering.

### B. Planar and Nonplanar Points

A point $\mathbf{p} \in \mathbf{P}$ can be locally described as lying on a 0-D, 1-D, 2-D, or 3-D manifold. A point $\mathbf{p}$ is described as 0-D if it is isolated, i.e., it does not have any neighbors. Examples of such points could be returns from a small chimney, a part of a tree, or simply a noisy return. The neighborhood of a 1-D point should all be on a straight line. A point $\mathbf{p}$ is described as 2-D if it lies locally on a planar surface, or 3-D if it lies near roof edges, roof ridges, curved roof surface, or trees. In Fig. 2, the bright points, mostly at the ridges, are considered as 3-D, whereas the dark ones are considered as 2-D.

The normals of 0-D, 1-D, or 3-D points are either nonexistent (0-D) or ambiguous (1-D, 3-D). Therefore, they should be detected and excluded from the subsequent clustering operation. A dimensionality analysis can help identify such points. A 2-D point needs a linear combination of two mutually orthogonal vectors (also called basis vectors). Similarly, a 3-D point needs three mutually orthogonal basis vectors. Suppose the Voronoi

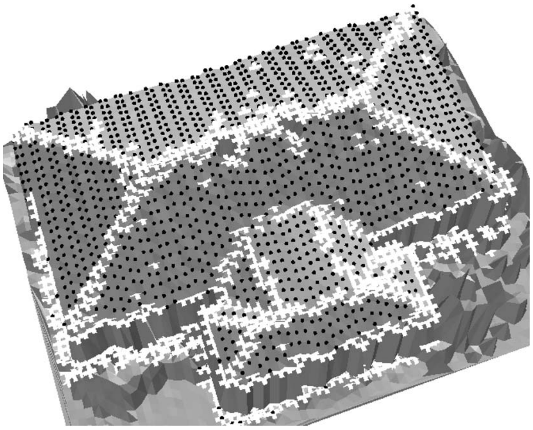


Fig. 2. (Dark dots) Two-dimensional (planar) points and (bright crosses) 3-D (nonplanar) points.

neighborhood $V_p$ consists of a set of $m_p$ points $\{\mathbf{X}_i\ (i = 1, \ldots, m_p)\}$. Let $\overline{\mathbf{X}}$ be their mean vector, then the covariance matrix $\mathbf{\Sigma_{XX}}$ is calculated as

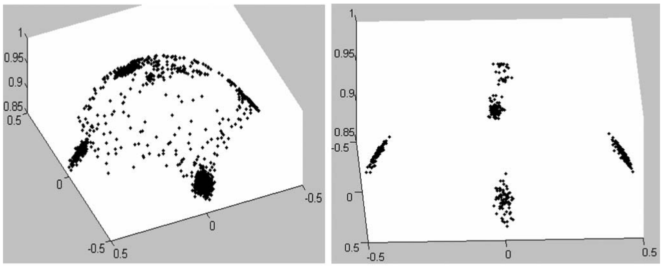
$$\mathbf{\Sigma_{XX}} = \sum_{i=1}^{m_p} (\mathbf{X}_i - \overline{\mathbf{X}})(\mathbf{X}_i - \overline{\mathbf{X}})^{\mathrm{T}}. \tag{1}$$

For this covariance matrix $\mathbf{\Sigma_{XX}}$, its eigenvalues $\lambda_1$, $\lambda_2$, $\lambda_3 (\lambda_1 \leq \lambda_2 \leq \lambda_3)$ and their corresponding eigenvectors $\mathbf{\Lambda}_1$, $\mathbf{\Lambda}_2$, $\mathbf{\Lambda}_3$ can be determined. The eigenvalues imply the dimensionality of the lidar point $\mathbf{p}$, while the eigenvectors represent the basis vectors of the neighborhood. If the dimensionality is two, i.e., if the vicinity of the point $\mathbf{p}$ lies on only one plane, there would exist only two nonzero eigenvalues. If the dimensionality of the point is three, all three eigenvalues will be nonzero. For a single isolated lidar point, there will be no neighborhood defined. It should be noted the aforementioned principle is similar to the principle component analysis or transform approach widely used in image processing [2] and machine learning [31].

Since lidar data sets have certain inherent noise associated with them, and taking into consideration factors such as surface roughness, it is not reasonable to expect a zero eigenvalue, even for a planar neighborhood. Therefore, if the normalized eigenvalue, which describes the local flattening at the lidar point

$$\overline{\lambda} = \lambda_1/(\lambda_1 + \lambda_2 + \lambda_3) \tag{2}$$

is less than a threshold $\varepsilon_{\overline{\lambda}}$, then the neighborhood can be hypothesized to be planar. The bright dots in Fig. 2 are determined to be points that satisfy the condition $\overline{\lambda} > \varepsilon_{\overline{\lambda}}$, where $\varepsilon_{\overline{\lambda}}$ is taken as 0.005 in this paper (see Section IV-A for discussion). Fig. 3(a) shows the distribution of the normals calculated for every lidar point of the same building. As can be noticed, the distribution is noisy with many randomly scattered normal vectors. Fig. 3(b) shows the distribution of the normals from only differentiable regions without including the breakline points. Such regions correspond to the dark points in Fig. 2. It is noticed that the distribution is less noisy, and the clusters of normals become apparent. This supports the statement that the breakline points should be detected and excluded from the subsequent clustering operation. The same building is shown in Fig. 5(a) and (b), while its detected breakline points and planar points are shown in Fig. 5(c) and (d), respectively.



(a) (b)

Fig. 3. Distribution of the surface normals of the building in Fig. 2. (a) For all points. (b) Without breakline points. The three coordinate axes are normal vector components.

## III. Roof Segmentation

Having detected and (temporarily) excluded the nonplanar points from consideration, we are left with the planar points. This section will cluster them into individual roof segments based on their normals determined in the previous section.

### A. *Clustering Methods*

Several types of clustering methods are available [31]–[33]. The hierarchical methods build a cluster hierarchy or a tree of clusters known as a dendrogram. Every cluster node contains child clusters, while sibling clusters partition the node of their common parent. The hierarchical algorithms can be agglomerative or divisive. The agglomerative algorithms are a bottom-up method that begins with the assumption that each data point (in feature space, also called feature vector, such as the normal vector in this paper) is a separate cluster and then merges some data points into one cluster based on a distance threshold. The process continues until the distance between clusters is larger than the given threshold. The divisive algorithms are a top-down method, which starts with a single cluster containing all data points and then successively splits the resulting clusters until the predefined number of clusters $k$ is reached. The partitioning methods iteratively relocate data points among various clusters until they achieve the greatest distinction. The popular $k$-means algorithm and its variations belong to this category. They produce $k$ different clusters of the greatest possible distinction [31] (see also the next section). The partition may also be formed using an objective function that leads to what are called decision boundaries. As an extension of the traditional partitioning method, the model-based methods assume that the data are generated from a mixture of probability distributions with each component corresponding to a different cluster. The goal of the clustering algorithm is then to maximize the overall probability or likelihood of the data [31], [33]. The density-based methods mostly work in the original data space [32], where the data are separated into different spatial regions. They are particularly useful when the shape and size of the clusters are random. These methods calculate the density at each data point and use a connectivity criterion to categorize all the "connected" data points with a similar density. Finally, it should be noted that all clustering methods work in a feature space, which can simply be the Cartesian coordinates $(x, y, z)$ (simply called segmentation in most applications) or the intensity values of multiple bands (image classification). The clustering algorithms group the data points close to each other and separate the data points far from each other. The definition of "close" and "far" depends on a distance measure, such as the Euclidean distance (in feature space) or its variations.

### B. Fuzzy $k$-Means Algorithm

This paper will use the fuzzy $k$-means algorithm [31], [33] to determine the clusters. In this method, a data point does not belong to only one cluster entirely. Instead, it can belong to any of the clusters at certain degree of belonging. The degree of belonging depends on the distance of a data point to a cluster center, such that the data point gets a higher weight for a close cluster center than for a far cluster center. The algorithm is similar to the statistically inclined expectation maximization (EM) algorithm [31]. Instead of using the probability densities that specify the degree to which a data point belongs to a cluster, the inverse distance measure is used in the fuzzy $k$-means algorithm. The algorithm works in the following manner.

1) Choose the approximate values for the number of clusters and the cluster centers.
2) Each data point (normal vector) is assigned a weight toward every cluster, depending on the distance of the data point to the cluster center.
3) A weighed mean is obtained for the location of each cluster center

$$\mathbf{C}^j = \sum_{i=1}^{m}\left(w_i^j \mathbf{N}_i\right) \Big/ \sum_{i=1}^{m} w_i^j \tag{3}$$

where

$$\mathbf{N}_i = (n_x \quad n_y \quad n_z)_i^{\mathrm{T}}. \tag{4}$$

4) Each data point is assigned to its closest cluster center.

In the aforementioned equations, $\mathbf{N}_i$ is the normal vector of the $i$th lidar point, i.e., the eigenvector corresponding to the minimum eigenvalue; $w_i^j$ is the weight of the $i$th data point toward the $j$th cluster; $m$ is the total number of data points; and $\mathbf{C}^j$ is the $j$th cluster center. The weights are also updated in iterations, depending on the distance between a data point and the newly calculated cluster centers. The steps 2, 3 and 4 are repeated until the cluster centers converge.

As a modification to the aforementioned fuzzy $k$-means approach, we introduce topologic information into the clustering. We argue that the clustering should depend not only on the distances to the cluster centers but also on the topologic properties, i.e., the dimensionality of the lidar points, such that a nonplanar data point should get a smaller weight toward the cluster center. Hence, in addition to the geometric Euclidean distances, we use $1/\overline{\lambda}$ as the topologic weight component to each data point. Thus, the combined weight or probability that a feature vector $\mathbf{N}_i$ belongs to the cluster center $\mathbf{C}^j$ is

$$w_i^j = P(\mathbf{N}_i \in \mathbf{C}^j) \propto \left(\frac{\lambda_1 + \lambda_2 + \lambda_3}{\lambda_1}\right)_i \Big/ \|\mathbf{N}_i - \mathbf{C}^j\|. \tag{5}$$

It should be noted that both the ratio $[(\lambda_1 + \lambda_2 + \lambda_3)/\lambda_1]_i$ and the distance $\|\mathbf{N}_i - \mathbf{C}^j\|$ are normalized to lie between zero and one, respectively, and the combined weight $[(\lambda_1 + \lambda_2 + \lambda_3)/\lambda_1]_i/\|\mathbf{N}_i - \mathbf{C}^j\|$ is also normalized to lie between zero and one. This technique of adjusting the traditional inverse distance weights to the data values reduces the effects of outliers to the fuzzy $k$-means (also called fuzzy $c$-means) classification, is therefore termed as robust fuzzy $c$-means method [34], [35].

### C. Cluster Centers and the Number of Clusters

The fuzzy $k$-means algorithm requires the number of clusters and their approximate locations to start the computation. In our case, this essentially means knowing the number of distinct planar directions and their approximate directions, neither of which is realistic. This section describes a method to determine the number of clusters and estimate their initial centers.

The potential-based clustering is introduced in [36]. "Potential" can be understood as the likelihood for a data point to be a cluster center. It first divides the feature space into grids and the potential of each grid is computed based on the distance from the grid center to the data points. A grid with many data points nearby will have a high potential, and the grid with the highest potential is chosen as the first cluster center. After that, the potential of the nearby grids is reduced based on their distance to the cluster center. This is to make sure that two grids that are close together do not become different clusters. The next cluster center is then chosen from the remaining grids with the highest potential. Reference [37] further develops this concept by using the actual data points instead of the grids as the cluster centers. The potential in [37] is analogous to the concept of energy potential, except that it is exponentially proportional to the inverse squared distance, with a fixed maximum distance representing the sphere of influence. The potential of a data point $\mathbf{N}_i$ is calculated as

$$p_i = \sum_{j=1}^{m} \exp\left\{-\frac{4}{r_a^2}\|\mathbf{N}_j - \mathbf{N}_i\|^2\right\} \tag{6}$$

where $m$ is the number of data points and $r_a$ is a positive constant. The constant $r_a$ defines a neighborhood, outside which the data points have little influence on the potential. As is shown in the aforementioned equation, the potential of a data point is dependent on its distances to all data points. A data point with many neighboring data points will have a high potential value.

The aforementioned potential concept is used to determine the cluster centers for the $k$-means clustering. The first cluster center is chosen as the data point that has the greatest potential (i.e., the maximum value of $p_i$ for all $i$). To reduce the possibility of the next cluster center being close to the first cluster center, the potential for every data point is then reduced as a function of its distance to the first cluster center

$$p_{i\_\mathrm{new}} = p_{i\_\mathrm{old}} - p_1^* \cdot \exp\left\{-\frac{4}{r_b^2}\|\mathbf{N}_i - \mathbf{N}_1^*\|^2\right\} \tag{7}$$

where $p_{i\_\mathrm{new}}$ and $p_{i\_\mathrm{old}}$ are, respectively, the new and old potentials of the data point $i$, $\mathbf{N}_1^*$ and $p_1^*$ are, respectively, the selected first cluster center and its potential. For this reason, this

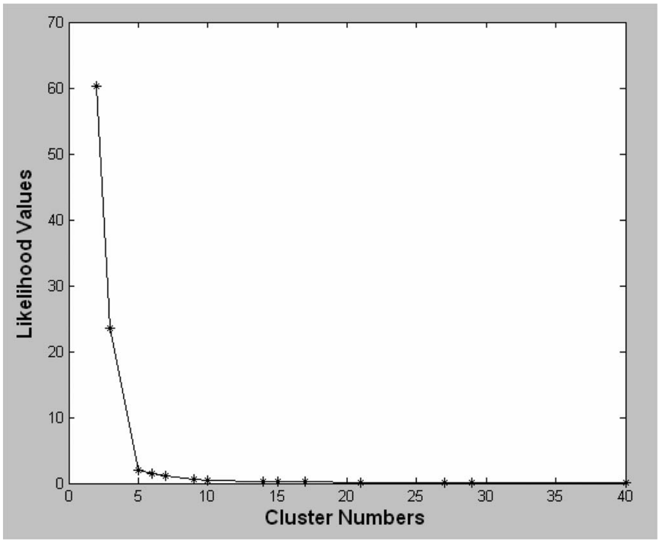


Fig. 4. Likelihood versus the number of clusters. The elbow point (5) is selected as the number of clusters. The plot is for the building in Fig. 2.

methodology is also called subtractive clustering. Reference [37] recommended a ratio of 1.5 between $r_b$ and $r_a$, such that the data points near the first cluster center will have greatly reduced potentials, and therefore will unlikely be selected as the next cluster center. The selection of the ratio depends on how close we want the clusters to be. In terms of building roof segmentation, the ratio determines how a roof segment can be directionally close to the other. Once the potentials of all the data points are updated based on (7), the data point with the current largest potential will be selected as the second cluster center. As in the process described by [37], the aforementioned process of acquiring new clusters stops when the cumulative potential in (7) reaches a threshold that is below 10% of the first potential.

It is clear that the value of the radius $r_a$ is critical to the clustering results. A smaller $r_a$ will yield a higher number of clusters and vice versa. Since we are operating in feature space, it is difficult to design a reasonable value for $r_a$. To overcome this problem, the method described in [37] is iteratively implemented. Starting from a small radius and increasing it gradually, fewer and fewer cluster centers will be obtained. The cluster centers generated for each $r_a$ are used as the input to the fuzzy $k$-means clustering algorithm. We choose the initial value of $r_a$ to be the mean value of the least 10% of the distances between data points.

At the end of the clustering, a likelihood estimate for each cluster is produced to measure the compactness of the cluster. Since a small value of $r_a$ produces more clusters than a large value of $r_a$, we can plot the likelihood estimates versus the number of clusters (instead of $r_a$), as shown in Fig. 4. We define the likelihood estimates as the average distance from the cluster center to the data points. Considering a case of $k$ clusters, the mean distance of data points from their respective cluster centers is calculated as

$$d^j = \sum_{i=1}^{m^j} d_i^j \Big/ m^j \tag{8}$$

where $d_i^j$ is the distance of a data point $i$ in cluster $j$ to its respective cluster center, $m^j$ is the number of data points in the $j$th cluster, $d^j$ is the mean distance of the $j$th cluster. The likelihood value for one clustering run is then defined as

$$d = \sum_{j=1}^{k} d^j \Big/ k. \tag{9}$$

As shown in Fig. 4, the likelihood falls sharply onto a place, after which the decrease becomes stable. The "elbow joint" is considered to be a good estimate on the number of clusters in the data set [31], [38]. At the end of this process, we know: 1) the number of clusters; 2) the cluster centers; and 3) the mapping of each data point to a cluster. Fig. 5(e) shows the clustering result at this stage.

### D. Separation of Parallel and Coplanar Planes

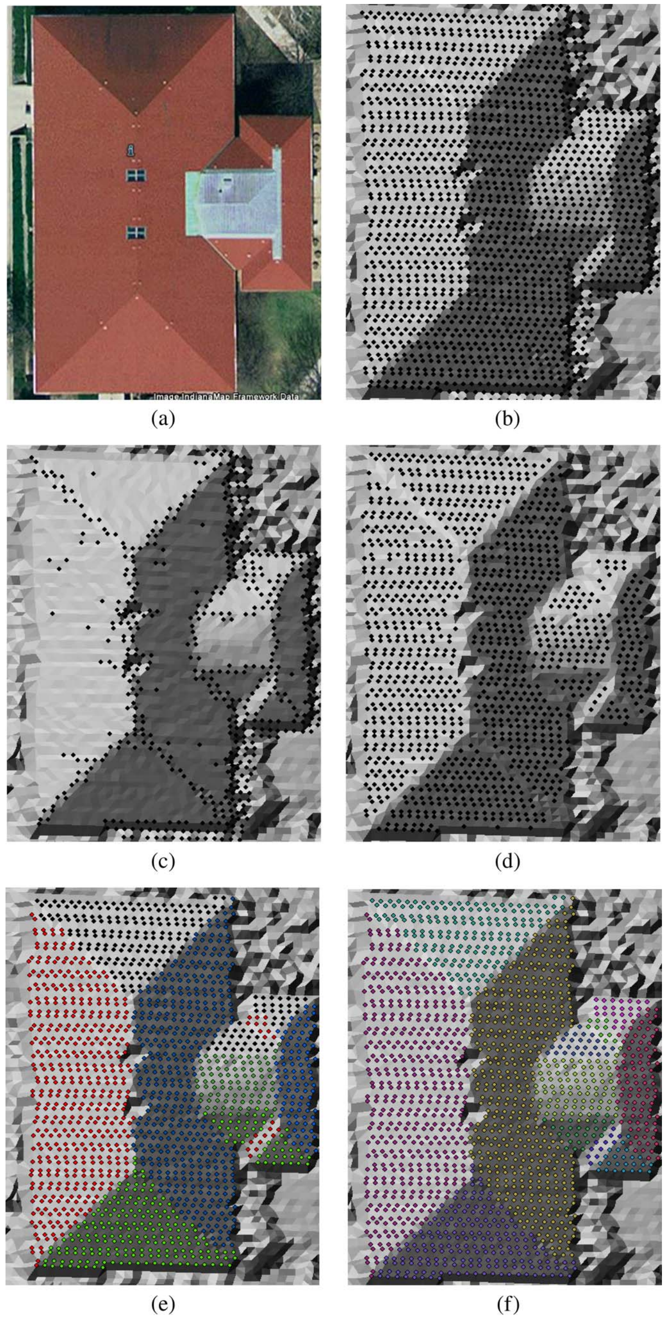
At the end of the aforementioned clustering process, we have the directions of all the roof planes, described by their normal vectors $\mathbf{N} = (n_x, n_y, n_z)^{\mathrm{T}}$. Each cluster actually describes a family of parallel planes with the equation $n_x x + n_y y + n_z z - \rho = 0$. The value of $\rho$ can be used to further segment the parallel planes. This is accomplished by plugging the lidar point coordinates of the same cluster into this equation. Lidar points that yield similar (within ±1.5 m) $\rho$ values are regarded to belonging to the same planar segment.

In the next step, we can attribute the initially detected nonplanar points (breakline points) back to their respective planes. Each breakline point is tested against every plane equation $n_x x + n_y y + n_z z - \rho = 0$. The plane that returns the minimum disclosure that is within a certain threshold (2.0 m) will be assigned to the lidar point being tested. The lidar points that do not belong to any plane are regarded as trees or ground and discarded from further processing.

Finally, a building roof can have two or more planar segments that are mathematically the same, but spatially separated. Such coplanar segments can be separated in the original data space based on the concept of density clustering and connectivity analysis [32]. The corresponding threshold in this paper is chosen as two times the lidar data ground spacing.

Fig. 5 shows the series of steps that have been discussed above for the example building shown in Fig. 2. The data set has a ground spacing about 1.1 m over the Purdue campus at West Lafayette, IN.

From left to right in the top row of Fig. 5, they are the building aerial image (from Google Earth, (a), and lidar points atop the triangulated irregular network (TIN) model (b). The second row displays the separated breakline points (c) and the planar points (d). It is clear that most breakline points are detected correctly. Returns from nonplanar points, such as small roof attachments are also well detected as breakline points. However, a closer look suggests that the separation is not always perfect as there exists "overremoval" of planar points. This usually should not be a concern because of the large number of redundant lidar points on the roof. Moreover, the mistakenly detected breakline points will be back-projected to the planar roofs



(a) (b) (c) (d) (e) (f)

Fig. 5. Clustering and segmentation of roof lidar points. (a) Image (from Google Earth). (b) Lidar points. (c) Breakline points. (d) Planar points. (e) Clustered points. (f) Planar segments.

after clustering. The opposite situation, "underremoval," occurs when real breakline points are not correctly detected. This takes place at relatively flat or smoothed roof ridges or edges where breaklines are wide and form a small flat region. Both mistakes may potentially cause inaccurate determination of roof plane boundaries in the subsequent steps. In this example, we have five clustered directional planes (e) shown with different colors. After separating parallel planes based on their distance from the coordinate origin, and separating coplanar segments based on density connectivity, we obtain the final segmented results, where the lidar points are segmented into 12 individual planar segments as color coded in (f).

## IV. Discussion and Evaluation

### A. *Discussion*

The selection of the neighborhood affects the calculation of the normals and hence the segmentation results. The Voronoi neighborhood is a natural selection since it does not need to specify how many lidar points to be selected and fully depends on the geometry of the data set. Unless the lidar point under consideration is at the edge of a building, at least five well-distributed neighboring points are likely to be included. This can provide a good estimation on the surface normals. As the current lidar systems can collect data in a resolution up to several or tens of points per square meter, essentially most roof planes can be segmented. Similar to many other neighborhood-based methods, however, some Voronoi neighbors of an interest point at the building or segment boundary can be far away. Including such points can lead to an incorrect normal estimate. Hence, neighboring points beyond a threshold of three times the lidar data ground spacing are excluded from normal vector calculation.

The breakline test needs a threshold for the normalized eigenvalue or local flattening $\overline{\lambda} = \lambda_1/(\lambda_1 + \lambda_2 + \lambda_3)$. A lidar point with $\overline{\lambda} \leq \varepsilon_{\overline{\lambda}}$ will be considered as a planar point. A small threshold may result in a planar region being classified as breaklines, while a large one will result in the opposite. Our experiments with different buildings found that $\varepsilon_{\overline{\lambda}} = 0.005$ gives satisfactory results. It can balance the two types of errors, namely, breakline points being clustered as planar (underremoval) and planar points being categorized as breaklines (overremoval). A slight bias toward the second type of error, i.e., a smaller threshold, does not significantly affect the final results since the high density of lidar data usually provides sufficient redundancy to determine a plane and the initially removed breakline points will ultimately be mapped back to their corresponding planes after clustering.

The aforementioned empirical selection of the threshold $\varepsilon_{\overline{\lambda}}$ can also be discussed theoretically. When fitting a plane over the neighborhood $V_p$ of a point $\mathbf{p}$ with the least squares method, it is shown in [39] that the smallest eigenvalue $\lambda_1$ is the sum of the squared distances of the points in $V_p$ to the least squares plane. In addition, the sum of the eigenvalues represents $\sum_{i=1}^{m_p}(x_i - \overline{x})^2 + (y_i - \overline{y})^2 + (z_i - \overline{z})^2$, with $m_p$ being the number of points in $V_p$. Thus, $\overline{\lambda}$ can be written as

$$\overline{\lambda} = \frac{\sum\limits_{i=1}^{m_p}(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 + (z_i - \hat{z}_i)^2}{\sum\limits_{i=1}^{m_p}(x_i - \overline{x})^2 + (y_i - \overline{y})^2 + (z_i - \overline{z})^2} \tag{10}$$

where $(\hat{x}_i, \hat{y}_i, \hat{z}_i)$ represents the least squares plane projection of point $i$, and $(\overline{x}, \overline{y}, \overline{z})$ is the center of $V_p$. Consider a 3 × 3 point distribution at a regular spacing $\Delta$, the sum of the squared distances to the centroid is $12\Delta^2$. If the uncertainty in the position of each lidar point is assumed to be 0.1 m, the numerator of (10) becomes $9 \times 0.01\ \mathrm{m}^2$. Let the lidar point spacing be $\Delta = 1.1$ m, this ratio turns out to be 0.006.

As many other pattern recognition problems, the clustering for segmentation is affected by the presence of noise. Fig. 3(a) shows a plot of the normal vector values, where each dot is a feature vector that represents the estimated normal vector at a lidar point. As shown, the dots are scattered in a rather random manner without a clear pattern. Directly working with these data would be very difficult to produce good clustering results. However, a pattern is apparent in Fig. 3(b) once the breakline points are excluded. The dots are now located in distinct clusters, indicating that there are only several planes facing different directions in the building. This justifies our strategy of detecting the breaklines and removing them from clustering due to their ambiguity. It is noticed that some dots seem to be slightly away from the cluster centers, which is most likely caused by lidar points at the breaklines or other nonplanar features that have not been completely removed when separating the planar and the nonplanar regions. The existence of some isolated dots away from the cluster centers implies that they are likely trees or other nonplanar points.

The potential-based clustering approach can reduce the effect of such off-center outliers that are mixed in the data points, and correctly determine the number and locations of these clusters. In the elbow-joint graph of Fig. 4 that plots the likelihood values for different number of clusters, we choose the number of clusters as five (5) at the place where the distinct change occurs. This way, the clustering process is able to determine the number of distinct planes with different normals and the lidar points of each directional plane.

The segmentation algorithms reviewed in Section I do not necessarily take into account the diversity of the geospatial nature of the lidar data sets, such as returns from trees, vertical walls, chimney, curved structures, and other irregular objects, in addition to the noise present in the data. Most segmentation algorithms assume a parametric nature for the data, which is definitely not the case with lidar data over an urban environment. We present a way to detect such points to achieve a robust segmentation. The eigenvalue analysis generates a measure for the planarity in the neighborhood of a lidar point. Thus, nonplanar points, such as trees or regions of breaklines at the intersection of two planes, are initially excluded from the subsequent clustering process. This avoids the general pitfalls of most region growing processes, which do not work well near edges. The reconstruction process then explicitly determines the location and parameters of the edges by using the segmented planes.

The aforementioned clustering process also features several improvements. It can select the number of clusters and a good approximation to the cluster centers. Since the fuzzy $k$-means algorithm is known to be sensitive to outliers in the data, we consider the effect of noise and remaining nonplanar normal vectors by further introducing the concept of topologic weight and using the least eigenvalue as a weighting factor, which provides a robust means for estimating planar directions.

It should be noted that the presence of a curved surface (e.g., a sport dome) in a building can lead to over segmentation. In this scenario, the lidar points of a curved surface will be segmented into more than one planar piece within certain tolerance. Some surface fitting or modeling steps may be followed based on the segmentation results. Since this paper assumes that all the building roofs are polyhedral, this scenario will be discussed elsewhere.

TABLE I
RESULTS (IN DEGREES) OF CLUSTERING DIRECTIONAL PLANES*

| | Cluster# | Count | Min | Max | Mean | $\sigma$ |
|---|---|---|---|---|---|---|
| within cluster | 1 | 403 | 0.050 | 12.588 | 1.768 | 1.262 |
| | 2 | 388 | 0.100 | 11.527 | 1.670 | 1.342 |
| | 3 | 160 | 0.119 | 12.612 | 2.257 | 2.305 |
| | 4 | 174 | 0.141 | 8.356 | 1.722 | 1.510 |
| | 5 | 78 | 0.854 | 13.029 | 5.359 | 2.573 |
| between-cluster distances for the five clustered planes | | | 24.537 | 52.508 | 35.319 | 10.106 |

*Cluster#: sequential cluster number; Count: the number of normal vectors in the cluster; Min, Max and Mean: the minimum, maximum and mean angular difference (in degrees) from the mean of one cluster (for within-cluster) or entire clusters (for between-clusters); $\sigma$ : standard deviation (in degrees) of the angular difference from the mean of one cluster or entire clusters.

TABLE II
PARAMETERS AND STATISTICS FOR SEGMENTED PLANES*

| Pl# | #Pts | $n_x$ | $n_y$ | $n_z$ | $\rho$ | Min | Max | $\sigma$ |
|---|---|---|---|---|---|---|---|---|
| 1 | 434 | 0.452 | 0.003 | 0.892 | 96.43 | -0.13 | 0.12 | 0.04 |
| 2 | 93 | 0.442 | 0.004 | 0.897 | 110.53 | -0.10 | 0.09 | 0.04 |
| 3 | 23 | -0.441 | -0.003 | 0.898 | -77.76 | -0.07 | 0.09 | 0.04 |
| 4 | 474 | -0.446 | -0.003 | 0.895 | -35.99 | -0.13 | 0.12 | 0.05 |
| 5 | 35 | -0.001 | 0.430 | 0.903 | 102.18 | -0.14 | 0.56 | 0.07 |
| 6 | 26 | -0.005 | 0.444 | 0.896 | 108.46 | -0.06 | 0.06 | 0.03 |
| 7 | 177 | -0.003 | 0.447 | 0.894 | 130.38 | -0.10 | 0.09 | 0.04 |
| 8 | 29 | 0.001 | -0.430 | 0.903 | -57.83 | -0.13 | 0.11 | 0.05 |
| 9 | 23 | 0.007 | 0.431 | 0.902 | 55.61 | -0.06 | 0.05 | 0.03 |
| 10 | 188 | 0.006 | -0.451 | 0.892 | -35.21 | -0.09 | 0.10 | 0.03 |
| 11 | 80 | 0.019 | -0.012 | 1.000 | 20.50 | -0.60 | 0.51 | 0.09 |
| UC | 19 | - | - | - | - | - | - | - |

* Pl#: sequential plane number; #Pts: the number of lidar points in a plane; $n_x$, $n_y$, $n_z$ and $\rho$ : plane parameters of directional consines and intercept (in meters); Min, Max and $\sigma$: min., max. and standard deviation of the distances of lidar points to their corresponding planes (in meters). UC: Unclassified.

## B. Evaluation

Three tables are produced to assess the performance of the aforementioned steps. The first table (Table I) captures the results from the clustering process. It shows that the normals in Fig. 3 are clustered into five (5) classes. It also lists the within-cluster minimum, maximum, mean, and standard deviation of the differences (in degrees) for the normal vectors with respect to their corresponding cluster centers. The maximum value of 2.6° for the standard deviation suggests that all normal vectors within this range of each other are considered to belong to the same cluster. Finally, the minimum separation of 24.5° between different clusters indicates the normals are very well separated.

Table II presents an evaluation of two counts: the assumption of polyhedral roof and the correct determination of the planar parameters. Once the roof points are segmented, their perpendicular distances $\rho_d = n_x x + n_y y + n_z z$ (with $n_x^2 + n_y^2 + n_z^2 = 1$) to the planes on which they are supposed to lie are evaluated. The columns of Min and Max indicate the signed minimum and maximum distances of a segmented point from its corresponding plane. The extreme values of Min and Max presented in the table can help draw several conclusions. It

TABLE III
STATISTICS (IN METERS) OF PLANAR SEGMENT COMPACTNESS*

| Sgmt# | #Pts | Min | Max | Mean | $\sigma$ |
|---|---|---|---|---|---|
| 1 | 474 | 0.58 | 1.47 | 1.16 | 0.23 |
| 2 | 434 | 0.70 | 1.37 | 1.14 | 0.15 |
| 3 | 177 | 0.69 | 1.50 | 1.10 | 0.19 |
| 4 | 188 | 0.69 | 1.41 | 1.12 | 0.15 |
| 5 | 80 | 0.79 | 1.29 | 1.08 | 0.12 |
| 6 | 93 | 0.86 | 1.39 | 1.13 | 0.14 |
| 7 | 27 | 1.01 | 1.31 | 1.14 | 0.07 |
| 8 | 34 | 0.94 | 1.32 | 1.14 | 0.08 |
| 9 | 24 | 0.95 | 1.50 | 1.19 | 0.19 |
| 10 | 21 | 1.04 | 1.36 | 1.17 | 0.09 |
| 11 | 10 | 1.00 | 1.28 | 1.16 | 0.10 |
| 12 | 13 | 0.87 | 1.32 | 1.13 | 0.15 |
| UC | 26 | - | - | - | - |

*Sgmt#: segment number; #Pts: the number of lidar points in a segment; Min, Max and $\sigma$: minimum, maximum and standard deviation of the triangle edge lengths in the triangulation of lidar points, all in meters.

shows the capability to separate two parallel planes. An extreme value of $-0.60$ m is obtained for this building. This indicates the worst nonplanarity within one planar cluster. The small standard deviations (max. 0.09 m) suggest the overall planarity of the segmented roof points. Together with the Min and Max columns, the standard deviation values measure the quality of roof segmentation. A small measure means that the corresponding points fit well with their planes, while a large value can indicate two things: the segmentation is not proper, i.e., the points may belong to different planes, or the assumption of the polyhedral nature of the roof is not valid. The unclassified (UC) row shows the results for points that do not match any of the planes generated from the process. A majority of these points lie inside a twin depression on the roof of the building (Fig. 5).

Table III presents a measure of compactness of each planar segment. The Pl# 3 in Table II is separated into two segments (Sgmt# 11 and 12) in Table III. The points of each segment are triangulated and the edge lengths of each triangle are recorded. Shown in Table III are the statistics of the triangle edge lengths for all the roof segments. The table shows the compactness of the resultant roof segments. It is shown that the average edge length in a roof segment is very close to the lidar point density (~1.1 m), and the small standard deviation of the lengths suggests that the triangles are mostly equilateral, which reflects the distribution of lidar points on a building roof. This shows that the final segmentation is density compact, and there is no apparent outlier in the results. The row labeled UC refers to the lidar points that are not classified and are mostly returns from vertical sections or walls of the roof. It has been included for the sake of completeness. It should be noted that the reason the count of UC points has increased from 19 in Table II to 26 in Table III is due to the fact that spatial separation of planar points has resulted in some isolated points (less than three in numbers) and hence they are added into the UC class.

TABLE IV
PLANE CLUSTERING STATISTICS (IN METERS) FOR 15 BUILDINGS*

| Building Name | #Pts | #UC | #Pls | Min | Max | $\sigma$ |
|---|---|---|---|---|---|---|
| Lambert | 6285 | 286 | 8 | -1.01 | 0.67 | 0.15 |
| Compt. Sci. | 1601 | 19 | 11 | -0.60 | 0.56 | 0.04 |
| Mech. Eng. | 3017 | 214 | 14 | -1.96 | 1.40 | 0.16 |
| Ag. 1 | 3226 | 271 | 15 | -1.29 | 1.21 | 0.24 |
| Stanley | 2128 | 59 | 14 | -1.05 | 1.28 | 0.11 |
| Chem. Eng. | 1774 | 56 | 10 | -0.88 | 0.79 | 0.10 |
| ISS | 2208 | 170 | 21 | -1.52 | 1.53 | 0.15 |
| Ag. 3 | 1515 | 49 | 13 | -0.97 | 1.40 | 0.17 |
| Dorm | 763 | 85 | 7 | -1.19 | 1.30 | 0.13 |
| Mathews | 1584 | 36 | 7 | -1.13 | 1.25 | 0.13 |
| Health Ctr | 2009 | 55 | 12 | -1.10 | 1.30 | 0.15 |
| Tourism | 2065 | 104 | 8 | -0.95 | 1.40 | 0.19 |
| Physics | 8942 | 302 | 22 | -1.72 | 1.14 | 0.17 |
| Stewart Ctr | 14748 | 1011 | 46 | -1.24 | 0.96 | 0.18 |
| Hovde | 1555 | 77 | 10 | -1.44 | 1.03 | 0.10 |
| Mean | | | | -1.20 | 1.14 | 0.15 |

- #UC: the number of unclassified lidar points

Finally, Table IV provides a summary statistics for 15 buildings over the Purdue University campus. Similar to Table II, Table IV shows that the average resolution among parallel planes across these buildings is ~1.2 m. The UC points represent lidar returns from vertical walls, small chimneys, overhanging trees, etc., that have not been classified into any roof planes. The ratio of UC points to the total number of points is ~5%. This is calculated by dividing the sum of the numbers presented in the column "UC" to the sum of the numbers presented in the column "#Pts." The UC point percentage also depends on the type of the roof. A roof with large number of small planes is likely to have more UC points.

## V. BUILDING RECONSTRUCTION

Building reconstruction is the determination of the building shape with minimum parameters or minimum geometric primitives (such as vertices) and their connectivity. This section uses the segmented lidar points to reconstruct the building roof and evaluate its quality.

### A. Method

The roof consists of roof planes, breaklines (ridges), edges (roof boundary), and the corresponding vertices. Since the clustering results have determined each planar segment and its belonging lidar points, the reconstruction step will intersect these planar segments to determine the roof breaklines and vertices.

An adjacency matrix is used to describe the connectivity among the planar segments. We define the distance of two planar segments as the minimum distance of all possible point combinations between the two segments

$$d(P,Q) = \min\left(d(p_i, q_j)\right) \quad \forall p_i \in P;\ \forall q_j \in Q \tag{11}$$

where $d(P,Q)$ is the distance between two planar segments $P$ and $Q$, $d(p_i, q_j)$ stands for the distance between any two points $p_i$ and $q_j$, respectively, in the segments $P$ and $Q$. As the

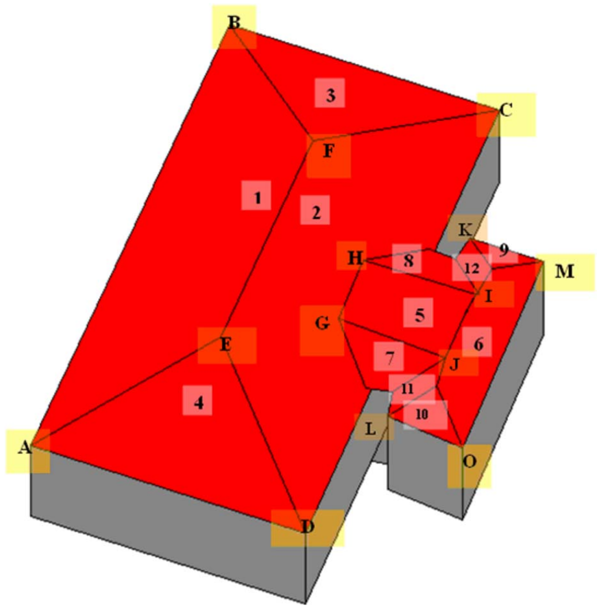


Fig. 6. Planar segment adjacency and vertex determination for building reconstruction.

TABLE V
Adjacency Matrix of the Building in Fig. 6

| Vertex | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | | * | * | * | | | | | | | | |
| 2 | * | | * | * | * | | * | * | | | | |
| 3 | * | * | | | | | | | | | | |
| 4 | * | * | | | | | | | | | | |
| 5 | | * | | | | * | * | * | | | * | * |
| 6 | | | | | * | | * | * | * | * | * | * |
| 7 | | * | | | * | * | | | | | * | |
| 8 | | * | | | * | * | | | | | | * |
| 9 | | | | | | * | | | | | | * |
| 10 | | | | | | * | | | | | * | |
| 11 | | | | | * | * | * | | | * | | |
| 12 | | | | | * | * | | * | * | | | |

result of such neighborhood evaluation, the adjacency matrix is formed for all the planar segments of a building. It represents the mutual adjacency between any two roof segments. For the building model shown in Fig. 6 (the same building as Figs. 2 and 5), the corresponding adjacency matrix is listed in Table V. For plane 1, the adjacency matrix suggests it is adjacent to planes 2, 3, and 4. The intersection of planes 1 and 2 is the breakline EF, with E and F denoting vertices. To determine these two vertices, we need the planes that are adjacent to both planes 1 and 2. We determine this from the adjacency matrix by looking at the rows of planes 1 and 2, and determining that planes 3 and 4 are adjacent to both planes 1 and 2. Under the assumption that the intersecting planes do not form a volume, i.e., we are dealing with a surface, the intersection of the three adjacent planes $\{1, 2, 3\}$ determines the vertex F, whereas E is obtained as the intersection of planes $\{1, 2, 4\}$. In general, a vertex is called degree $r$-vertex if it is the intersection of $r$ planes. As shown in Fig. 6, E and F are degree 3 vertices and can be, respectively, determined by solving the equations of the three corresponding planes. Similarly, vertex I and J are determined, respectively, by using four planar equations because their degree is 4.

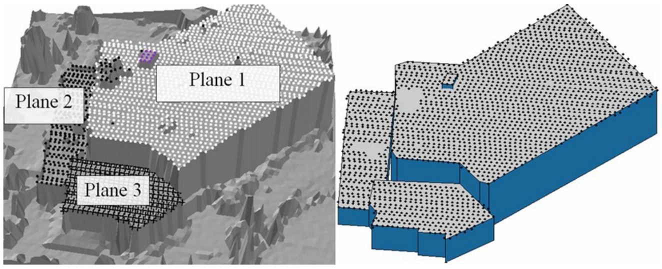


Fig. 7. (Left) Building with flat roof planes and (right) its reconstruction.

However, singularity occurs when determining a vertex with $r < 3$. For vertices A and B on the building edge or boundary, $r$ is two. In this situation, we need to introduce the boundary constraint. Vertices A and B are, respectively, forced to lie on the breaklines AE and BF determined by using the adjacency matrix. The actual location of B is determined by projecting all the boundary points in planes 1 and 3 that are "close" (i.e., points that lie at a distance slightly less than twice of the point spacing) to the line BF and then selecting the end point. An alternative solution is to introduce one or more vertical planes that pass through the building edges [26]. For example, to determine the vertex B, two vertical planes, respectively, passing the boundary line BA and BC are introduced. They will join the roof planes 1 and 3 to determine the vertex B. Another singular situation is that no adjacent plane $(r = 1)$ is found for a given roof segment. This occurs frequently in buildings with flat roofs. As shown in Fig. 7, the building has three major roof planes, none of which has adjacent planes. Hence, such buildings can only be reconstructed by regularizing their boundaries as discussed in [40].

This section presents a general solution to the aforementioned boundary reconstruction problem. It is extended from the authors' recent work [40], which determines a building's footprint through a local convex hull approach and then regularizes it to two mutually orthogonal directions. This process [40] first approximates the roof boundary from the segmented lidar points by applying a local convex hull approach. In the next step, the longer line segments are determined from the approximate boundary, which estimates the orientation of the building footprint. All line segments are then constrained by using a least squares formulation to lie perpendicular or parallel to each other. The aforementioned process is generalized as below to consider arbitrary directions. The central idea is to divide the boundary into several mutually perpendicular or parallel line groups; however, there is no constraint between the groups. We consider the boundary of a roof plane shown in Fig. 7. Let the equation of the plane be $n_x x + n_y y + n_z z - \rho = 0$, with $n_x^2 + n_y^2 + n_z^2 = 1$. The first step is to extract the longer edges of the roof plane. This is done by sequentially following the boundary points and looking for positions where the slopes of two consecutive edges are significantly different. Points on consecutive edges with similar slopes are grouped to one line segment. In this way, we will form a set of line segments $\{l_1, l_2, \ldots, l_u\}$, from which longer (>4 points) line segments $\{L_1, L_2, \ldots, L_v\}$ are then selected. The longer line segments are modeled by the equation $(x - \overline{x}_s)/a_s = (y - \overline{y}_s)/b_s = (z - \overline{z}_s)/c_s$, where $\{\bar{x}, \bar{y}, \bar{z}\}_s$ are the mean coordinates of the points in the long line $L_s$ The parameters of the line $\{a, b, c\}_s$ define the direction of the lines. In the previous work [40], these parameters are used to divide the line segments into two mutually orthogonal categories. In this paper, we introduce another level of division. For any long line segment $L_s$ with the parameters $\{a, b, c\}_s$, all line segments (given by $\{a, b, c\}_r$) that yield the dot product $\{a\hat{i} + b\hat{j} + c\hat{k}\}_s \bullet \{a\hat{i} + b\hat{j} + c\hat{k}\}_r = 0$ (perpendicular) or $a_s/a_r = b_s/b_r = c_s/c_r$ (parallel) are collected, and categorized into a category $C_1$. In a similar manner, we generate further categories $C_2, C_3, \ldots, C_n$ such that within each category, we have mutually parallel or perpendicular line segments. The line segments within each category are processed in a manner similar to what was expounded in [40]. All line segments in all categories are also constrained to lie on the roof plane $n_x x + n_y y + n_z z - \rho = 0$ by the vector dot product $\{a\hat{i} + b\hat{j} + c\hat{k}\}_s \bullet \{n_x\hat{i} + n_y\hat{j} + n_z\hat{k}\} = 0$ and with the additional constraint that points on the line should always be on the plane, i.e., satisfy the equation of the plane. This ensures that the least squares solution within each category consists of a series of mutually parallel and perpendicular lines that lie on the plane. However, there is no constraint among the line segments between different categories. This is regarded as a generalization of the previous work where a building was considered to have only two mutual parallel or perpendicular directions, whereas we do not restrict the number of orthogonal directions in the generalized framework in this paper.

In summary, a building is reconstructed based on the segmented planes and their adjacency matrix. A building consists of a collection of planar segments, which are defined by their planar equations and ordered boundary points. The adjacency matrix defines the connectivity among the planar segments. The intersection of three or more adjacent planes yields the roof vertices. A roof edge is formed by connecting two vertices where two constituent planes intersect. For flat roofs, they are simply reconstructed by boundary approximation and regularization.

### B. Evaluation

The aforementioned reconstruction results represent a vector or boundary model of the building roof. The accuracy of the derived roof planes depends on the quality of segmentation as the equations of the planes are estimated from the segmented points. Therefore, smaller planes with less number of segmented points are expected to be less accurate than the larger ones. Another reason for the expected drop in accuracy for smaller planes is that a higher percentage of the segmented points from a smaller plane are near its boundary. Table VI shows the estimated accuracy for the roof planes of the building shown in Fig. 6. The estimates are based on the assumption that the horizontal standard deviation $\sigma_x = \sigma_y = 30$ cm and the vertical standard deviation $\sigma_z = 10$ cm for each lidar point. All the lidar points of a planar segment are used to determine the plane equation based on the least squares criterion. Let the equation of the plane be $n_x x + n_y y + n_z z - \rho = 0$, with $n_x^2 + n_y^2 + n_z^2 = 1$, then its direction consines are $\cos(\alpha) = n_x$, $\cos(\beta) = n_y$, $\cos(\gamma) = n_z$. The quantities $\sigma_\alpha$, $\sigma_\beta$, and $\sigma_\gamma$ represent the standard deviations of the direction angles. As expected, larger planes have higher accuracy than smaller ones, with a range varying from 0.03° to 1.03°. It is noticed that the smaller planes demonstrate large and unstable fitting uncertainties, particularly in the intercept distance $\rho$.

TABLE VI
ESTIMATED ACCURACY OF PLANE FITTING FOR THE BUILDING IN FIG. 6

| Pl# | #Pts | $\sigma_\alpha$ | $\sigma_\beta$ | $\sigma_\gamma$ | $\sigma_\rho$ |
|---|---|---|---|---|---|
| | | degrees | | | meters |
| 1 | 474 | 0.033 | 0.012 | 0.033 | 0.19 |
| 2 | 434 | 0.038 | 0.012 | 0.038 | 0.15 |
| 3 | 177 | 0.042 | 0.064 | 0.064 | 0.36 |
| 4 | 188 | 0.043 | 0.061 | 0.061 | 0.16 |
| 5 | 80 | 0.057 | 0.072 | 0.337 | 0.20 |
| 6 | 93 | 0.247 | 0.062 | 0.247 | 0.75 |
| 7 | 27 | 0.290 | 0.504 | 0.504 | 1.75 |
| 8 | 34 | 0.362 | 0.525 | 0.526 | 3.06 |
| 9 | 24 | 0.874 | 0.694 | 0.780 | 4.91 |
| 10 | 21 | 0.595 | 0.932 | 0.931 | 3.16 |
| 11 | 10 | 0.645 | 0.886 | 0.887 | 5.89 |
| 12 | 13 | 1.034 | 0.733 | 0.954 | 7.58 |

Another concern in building reconstruction is the accuracy of the vertices generated. Most of the vertices have been determined either using the minimal required intersection of three planes, or as in case of the outer ones, using the assumption of vertical walls. Fig. 8 shows five reconstructed buildings along with their images and color-coded segmented lidar points. The breaklines, including roof edges and ridges, are represented in the reconstructed models. The lack of lidar returns from the vertical sections of the building prevents their participation in the adjacency matrix. Therefore, the vertices of a building are determined by essentially one of the three methods: Type A, intersection of lines, e.g., for flat roof; Type B, intersection of lines (equivalent to vertical walls) and planes; and Type C, intersection of planes. The vertices marked "A" in Fig. 8, correspond to vertices that are formed without any explicit intersection of planes. They are most easily seen as corners of roof planes that have no adjacent roofs, such as "flat roofs." Such vertices are the least accurate ones since they simply do not use planar extension and determined by intersecting two lines. The vertices marked "B" correspond to locations that are often formed by the intersection of two planes and two corresponding "vertical walls" (expressed as line equations). Since the vertical walls are essentially formulated by the segment boundaries, the participation of roof planes will enhance the accuracy of line intersection. As the result, Type B vertices are less affected by the intersecting lines and therefore have a better accuracy than Type A vertices. The vertices marked "C" are locations that are formed from the intersection of three or more planes. Most interior roof planes have this type of vertices and can therefore benefit from the areal extent of the lidar point clouds. This

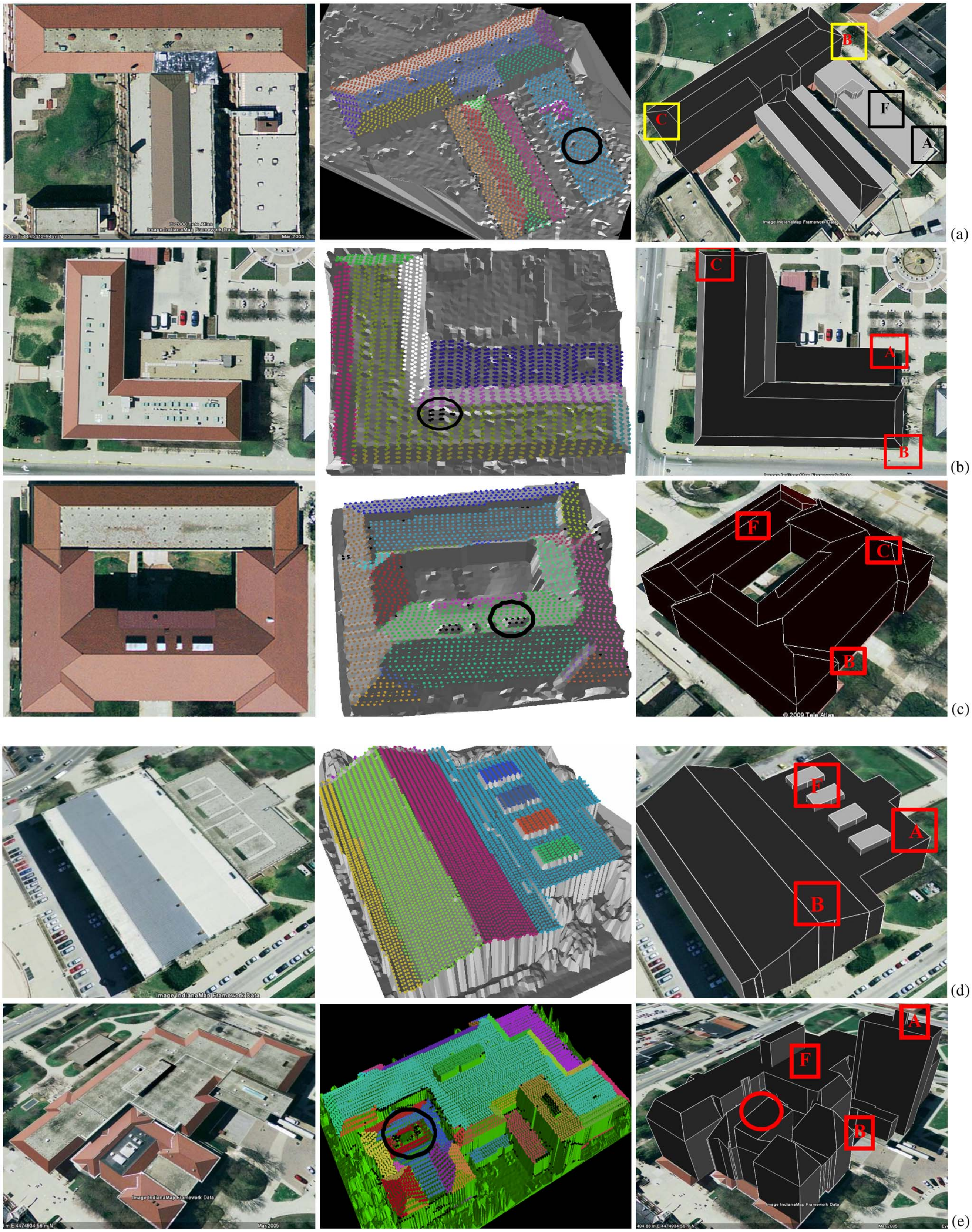

© 2008 Tele Atlas
Image IndianaMap Framework Data
Mar 2005
B
C
F
A
Image IndianaMap Framework Data
(a)
C
A
B
(b)
F
C
B
© 2009 Tele Atlas
(c)
Image IndianaMap Framework Data
F
A
B
(d)
Image IndianaMap Framework Data
Mar 2005
A
F
B
Image IndianaMap Framework Data
Mar 2005
(e)

Fig. 8. Building images, segmented lidar points, and reconstructed buildings.

makes Type C vertices have the best accuracy. Based on the aforementioned discussion, vertices are essentially determined as an intersection of lines (Type A) or planes (Type C) or their combinations (Type B). The error propagation law then allows us to estimate the average uncertainty of Type A, B, and C vertices be 0.19, 0.14, and 0.12 m, respectively. This uncertainly may increase up to 0.42 m for small roof segments, such as the ones in the building of Fig. 6 with only ten points. It should be noted that the actual range of the uncertainty for vertices may vary with lidar data sets, however, the aforementioned evaluation and understanding can lead to a general conclusion that building interior can be reconstructed with a better accuracy than its outer boundary. In addition, Fig. 8 uses an oval to show representative areas of unclassified lidar points, and "F" for representative flat roof planes determined by boundary tracing and regularization. Unclassified lidar points may cause wrong reconstruction, as shown in Fig. 8(e).

## VI. Conclusion

We have presented a framework to segment a building and reconstruct it from unstructured airborne lidar point clouds. A polyhedral model for the roof is assumed as this constitutes most buildings. The segmentation step starts with the detection of nonplanar points, such as the points on roof ridges and edges, and trees. This is necessary because of the ambiguity that they may belong to more than one plane or no plane at all. This process is similar to robust interpolation, where the interpolation process proceeds after the removal of outliers (e.g., RANSAC). It is also similar to the semisupervised classification process where the training data sets are compared with edge detection results in the image and those training pixels that lie on edges are removed. The eigenanalysis is proven to be a theoretically sound and practically effective technique that can satisfactorily detect these nonplanar points. The normalized eigenvalue $\overline{\lambda}$, also termed as local flattening, is essentially a measure of flatness in the vicinity of a point, and can serve as a critical value for nonplanar point detection. The Voronoi neighborhood is regarded as a natural neighborhood that does not need to specify the size of the neighborhood. It is shown to be suitable for surface normal estimation. However, care should be exercised in areas near the roof boundary, where the Voronoi neighbors may not be close together. Once the nonplanar points are excluded, finding the roof planar segments can be rather robustly accomplished through a clustering process. It is shown that the potential-based method can yield a stable estimate on the number of clusters and initial cluster centers, which are needed for the following fuzzy $k$-means clustering calculation. The concept of weight in the fuzzy $k$-means method is extended to consider both the geometric similarity and topologic similarity for a point to belong to a certain cluster. This way the points with a higher likelihood to be planar will get a higher weight toward the cluster center. The separation of parallel segments and coplanar segments can be fairly easily accomplished based on the planar equations and the connectivity, respectively. The entire segmentation process can be evaluated by three quality matrices, which are, respectively, for the separation of planes with distinct directions, separation of parallel planes, and separation of coplanar elements. It is shown the proposed solution framework can achieve an overall within-cluster precision of 2.5° and between-class separation as large as 25°. The distance separation between two parallel planes can be set based on the complexity of roof structures, while the final segments should each have a point density similar to the one of the input lidar points. The three matrices together represent the quality of planar roof segmentation from lidar data. The test results of 15 buildings with diverse complexity demonstrate the proposed framework and its implementation steps.

The reconstruction process consists of determining the location of the roof vertices, roof ridges, and roof edges or boundaries. The adjacency matrix of the aforementioned derived planar segments is essential for this purpose. Based on this matrix, roof vertices, i.e., the intersection of roof ridges or edges are determined by intersecting the involved planes, which include roof segments and possibly vertical walls or roof boundaries as the imposed constraints. Determining a vertex by intersecting planes other than lines strengthens its reliability and makes the best use of the dense lidar points distributed in a plane. Single roof planes without adjacent planes are determined directly by using their boundary representations. This way, the resultant buildings are topological consistent in terms of their constituent components: vertices, lines, and polygons, with all being embedded in a 3-D space. It is shown that roof interior can be reconstructed with a better accuracy than roof outlines. Moreover, the newly extended boundary regularization approach can handle building roofs of diverse complexities and produces topologically correct and geometrically accurate building models.

## References

[1] C. Brenner, "Building reconstruction from images and laser scanning," *Int. J. Appl. Earth Obs. Geoinf.*, vol. 6, no. 3/4, pp. 187–198, 2005.

[2] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*., 2nd ed. Englewood Cliffs, NJ: Prentice-Hall, 2002, pp. 567–641.

[3] T. Fan, G. Medioni, and R. Nevatia, "Segmented description of 3-D surfaces," *IEEE Trans. Robot. Autom.*, vol. RA-3, no. 6, pp. 527–538, Dec. 1987.

[4] J. Wang and J. Shan, "Segmentation of LiDAR point clouds for building extraction," in *Proc. Amer. Soc. Photogramm., Remote Sens. Annu. Conf.*, Baltimore, MD, Mar. 9–13, 2009.

[5] X. Y. Jiang and H. Bunke, "Fast segmentation of range images into planar regions by scan line grouping," *Mach. Vis. Appl.*, vol. 7, no. 2, pp. 115–122, Apr. 1994.

[6] F. Rottensteiner and C. Briese, "Automatic generation of building models from LiDAR data and the integration of aerial images," in *Proc. Int. Soc. Photogramm. Remote Sens.*, Dresden, Germany, 2003, vol. 34, pp. 174–180.

[7] A. Alharthy and J. Bethel, "Detailed building reconstruction from airborne laser data using a moving surface method," *Int. Arch. Photogramm. Remote Sens.*, vol. 35, no. B3, pp. 213–218, 2004.

[8] M. Peternell and T. Steiner, "Reconstruction of piecewise planar objects from point clouds," *Comput.-Aided Des.*, vol. 36, no. 4, pp. 333–342, 2004.

[9] F. Rottensteiner, J. Trinder, S. Clode, and K. Kubik, "Using the Dempster–Shafer method for the fusion of LIDAR data and multi-spectral images for building detection," *Inf. Fus.*, vol. 6, no. 4, pp. 283–300, 2005.

[10] G. Forlani, C. Nardinocchi, M. Scaiono, and P. Zingaretti, "Complete classification of raw LIDAR and 3D reconstruction of buildings," *Pattern Anal. Appl.*, vol. 8, no. 4, pp. 357–374, Feb. 2006.

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

[12] A. Bab-Hadiashar and N. Gheissari, "Range image segmentation using surface selection criterion," *IEEE Trans. Image Process.*, vol. 15, no. 7, pp. 2006–2018, Jul. 2006.

[13] H. Woo, E. Kang, S. Y. Wang, and K. H. Lee, "A new segmentation method for point cloud data," *Int. J. Mach. Tools Manuf.*, vol. 42, no. 2, pp. 167–178, 2002.

[14] J. Huang and C. Menq, "Automatic data segmentation for geometric feature extraction from unorganized 3-D coordinate points," *IEEE Trans. Robot. Autom.*, vol. 17, no. 3, pp. 268–279, Jun. 2001.

[15] R. O. Duda and P. E. Hart, "Use of the Hough transformation to detect lines and curves in pictures," *Commun. ACM*, vol. 15, no. 1, pp. 11–15, Jan. 1972.

[16] F. Tarsha-Kurdi, T. Landes, and P. Grussenmeyer, "Hough-transform and extended RANSAC algorithms for automatic detection of 3D building roof planes from LiDAR data," in *Proc. Int. Soc. Photogramm. Remote Sens.*, 2007, vol. 36, pp. 407–412.

[17] G. Vosselman and S. Dijkman, "3D building model reconstruction from point clouds and ground plans," *Int. Arch. Photogramm. Remote Sens.*, vol. 34, no. 3/W4, pp. 37–43, 2001.

[18] J. Overby, L. Bodum, E. Kjems, and P. M. Iisoe, "Automatic 3D building reconstruction from airborne laser scanning and cadastral data using Hough transform," *Int. Arch. Photogramm. Remote Sens.*, vol. 35, pt. B3, pp. 296–301, 2004.

[19] M. Fischler and R. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[20] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*., 2nd ed. Cambridge, U.K.: Cambridge Univ. Press, 2004, pp. 117–123.

[21] F. Tarsha-Kurdi, T. Landes, P. Grussenmeyer, and M. Koehl, "Model-driven and data-driven approaches using LIDAR data: Analysis and comparison," in *Proc. Photogramm. Image Anal.*, 2007, vol. 36, pp. 87–92.

[22] R. L. Hoffman and A. K. Jain, "Segmentation and classification of range images," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-9, no. 5, pp. 608–620, Sep. 1987.

[23] S. Filin and N. Pfeifer, "Segmentation of airborne data using a slope adaptive filter," *ISPRS J. Photogramm. Remote Sens.*, vol. 60, pp. 71–80, 2006.

[24] A. Nizar, S. Filin, and Y. Doytsher, "Reconstruction of buildings from airborne laser scanning data," in *Proc. Amer. Soc. Photogramm., Remote Sens. Annu. Conf.*, Reno, NV, 2006.

[25] A. Sampath and J. Shan, "Clustering based planar roof extraction from LiDAR data," in *Proc. Amer. Soc. Photogramm., Remote Sens. Annu. Conf.*, Reno, NV, 2006.

[26] J. Shan and A. Sampath, "Building extraction from LiDAR point clouds based on clustering techniques," in *Topographic Laser Ranging and Scanning: Principles and Processing*, J. Shan and C. Toth, Eds. Boca Raton, FL: CRC Press, 2008, ch. 15, pp. 423–446.

[27] Y. Liu and Y. L. Xiong, "Automatic segmentation of unorganized noisy point clouds based on the Gaussian map," *Comput.-Aided Des.*, vol. 40, no. 5, pp. 576–594, May 2008.

[28] D. Marr and E. Hildreth, "Theory of edge detection," *Proc. R. Soc. Lond., B*, vol. 207, no. 1167, pp. 187–217, Feb. 1980.

[29] R. Vaillant and O. Faugeras, "Using extremal boundaries for 3-D object modeling," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 14, no. 2, pp. 157–173, Feb. 1992.

[30] D. OuYang and H. Y. Feng, "On the normal vector estimation of point cloud from smooth surfaces," *Comput.-Aided Des.*, vol. 37, no. 10, pp. 1071–1079, 2005.

[31] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*., 2nd ed. Hoboken, NJ: Wiley, 2001, pp. 519–598.

[32] P. Berkhin, "A survey of clustering data mining techniques," in *Grouping Multidimensional Data: Recent Advances in Clustering*, J. Kogan, C. Nicholas, and M. Teboulle, Eds. New York: Springer-Verlag, 2006, pp. 25–71.

[33] A. K. Jain, M. N. Murty, and P. J. Flynn, "Data clustering: A review," *ACM Comput. Surv.*, vol. 31, no. 3, pp. 264–323, 1999.

[34] A. Banerjee and R. N. Dave, "The fuzzy mega-cluster: Robustifying FCM by scaling down memberships," in *Lecture Notes in Artificial Intelligence*, vol. 3613, L. Wang and Y. Jin, Eds. Berlin, Germany: Springer-Verlag, 2005, pp. 444–453.

[35] W. Caia, S. Chen, and D. Zhanga, "Fast and robust fuzzy c-means clustering algorithms incorporating local information for image segmentation," *Pattern Recognit.*, vol. 40, no. 3, pp. 825–838, 2007.

[36] R. Yager and D. Filev, "Approximate clustering via the mountain method," *IEEE Trans. Syst., Man, Cybern.*, vol. 24, no. 8, pp. 1279–1284, Aug. 1994.

[37] S. Chiu, "Fuzzy model identification based on cluster estimation," *J. Intell. Fuzzy Syst.*, vol. 2, no. 3, pp. 267–278, Sep. 1994.

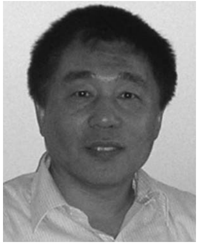
[38] R. Tibshirani, G. Walther, and T. Hastie, "Estimating the number of clusters in a data set via the gap statistic," *J. R. Stat. Soc., Ser. B Stat. Methodol.*, vol. 63, no. 2, pp. 411–423, 2001.

[39] A. Shuchat, "Generalized least squares and eigenvalues," *Amer. Math. Mon.*, vol. 92, no. 9, pp. 656–659, 1985.

[40] A. Sampath and J. Shan, "Building boundary tracing and regularization from airborne LiDAR point clouds," *Photogramm. Eng. Remote Sens.*, vol. 73, no. 7, pp. 805–812, 2007.

**Aparajithan Sampath** received the B.S. degree in geoinformatics engineering from Anna University, Chennai, India, and the M.S. degree in geomatics engineering from Purdue University, West Lafayette, IN, where he is currently working toward the Ph.D. degree in the School of Civil Engineering.

He is currently a Photogrammetry and Calibration/Validation Engineer with the Stinger Ghaffarian Technologies, Inc., U.S. Geological Survey's Center for Earth Resources Observation and Science, Sioux Falls, SD.



**Jie Shan** (M'08) received the Ph.D. degree in photogrammetry and remote sensing from Wuhan Technical University of Surveying and Mapping, Wuhan, China, in 1989.

He has held faculty positions with universities in China and Sweden, and has been a Research Fellow in Germany. He is currently an Associate Professor in geomatics engineering with the School of Civil Engineering, Purdue University, West Lafayette, IN. His area of interests includes sensor geometry, pattern recognition from images and lidar data, object extraction and reconstruction, urban remote sensing, and automated mapping.

Dr. Shan is the recipient of multiple academic awards, among which are the Talbert Abrams Grand Award, the Environmental Systems Research Institute Award for Best Scientific Paper in Geographic Information System (GIS) (First Place), and the Excellence in GIS Award. He has been the Cochair of the Remote Sensing Data Fusion Working Group and the Pattern Recognition for Remote Sensing Working Group, both in the International Society for Photogrammetry and Remote Sensing. He is an Associate Editor for IEEE Geoscience and Remote Sensing and the Assistant Editor for *Photogrammetric Engineering and Remote Sensing*. He is also a member of the American Society for Photogrammetry and Remote Sensing, and the American Geophysical Union.