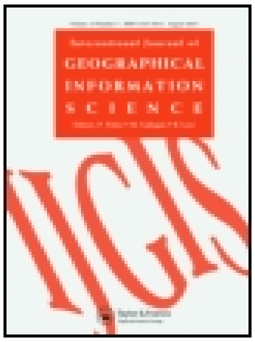

International Journal of Geographical Information Science

ISSN: 1365-8816 (Print) 1362-3087 (Online) Journal homepage: http://www.tandfonline.com/loi/tgis20

3D building roof reconstruction from airborne LiDAR point clouds: a framework based on a spatial database

Rujun Cao, Yongjun Zhang, Xinyi Liu & Zongze Zhao

To cite this article: Rujun Cao, Yongjun Zhang, Xinyi Liu & Zongze Zhao (2017): 3D building roof reconstruction from airborne LiDAR point clouds: a framework based on a spatial database, International Journal of Geographical Information Science, DOI: 10.1080/13658816.2017.1301456

To link to this article: http://dx.doi.org/10.1080/13658816.2017.1301456

Published online: 09 Mar 2017.

INTERNATIONAL JOURNAL OF GEOGRAPHICAL INFORMATION SCIENCE, 2017
http://dx.doi.org/10.1080/13658816.2017.1301456

Taylor & Francis
Taylor & Francis Group

ARTICLE

3D building roof reconstruction from airborne LiDAR point clouds: a framework based on a spatial database

Rujun Cao[a], Yongjun Zhang[a*], Xinyi Liu [a] and Zongze Zhao [b]

[a]School of Remote Sensing and Information Engineering, Wuhan University, Wuhan, China; [b]School of Surveying and Land Information Engineering, Henan Polytechnic University, Jiaozuo, China

ABSTRACT

Three-dimensional (3D) building models are essential for 3D Geographic Information Systems and play an important role in various urban management applications. Although several light detection and ranging (LiDAR) data-based reconstruction approaches have made significant advances toward the fully automatic generation of 3D building models, the process is still tedious and time-consuming, especially for massive point clouds. This paper introduces a new framework that utilizes a spatial database to achieve high performance via parallel computation for fully automatic 3D building roof reconstruction from airborne LiDAR data. The framework integrates data-driven and model-driven methods to produce building roof models of the primary structure with detailed features. The framework is composed of five major components: (1) a density-based clustering algorithm to segment individual buildings, (2) an improved boundary-tracing algorithm, (3) a hybrid method for segmenting planar patches that selects seed points in parameter space and grows the regions in spatial space, (4) a boundary regularization approach that considers outliers and (5) a method for reconstructing the topological and geometrical information of building roofs using the intersections of planar patches. The entire process is based on a spatial database, which has the following advantages: (a) managing and querying data efficiently, especially for millions of LiDAR points, (b) utilizing the spatial analysis functions provided by the system, reducing tedious and time-consuming computation, and (c) using parallel computing while reconstructing 3D building roof models, improving performance.

ARTICLE HISTORY
Received 16 April 2016
Accepted 28 February 2017

KEYWORDS
LiDAR point cloud; roof reconstruction; segmentation; spatial database; parallel computation

1. Introduction

Three-dimensional (3D) building models are essential for 3D Geographic Information Systems (GIS) and play an important role in numerous urban management applications. Since the data acquired by Light Detection and Ranging (LiDAR) systems are dense, accurate and geo-referenced in 3D (Leberl *et al*. 2010, Toth and Jóźków 2016), they are often used to reconstruct building models (Haala and Kada 2010, Rottensteiner *et al*. 2014). Polyhedral buildings are common in urban areas, and polyhedral models, which

CONTACT Yongjun Zhang zhangyj@whu.edu.cn
*Present address: School of Remote Sensing and Information Engineering, Wuhan University, Wuhan, China

represent buildings with simple planar patches, are sufficient for such applications (Sampath and Shan 2010, Xu *et al.* 2014). These buildings can be reconstructed in a fully automatic manner. However, the process for automatically reconstructing building models from point clouds remains challenging. The major difficulties include the absence of local data and noisy data or data that are inefficient for the reconstruction of sophisticated building structures. The accuracy of building boundaries and detailed roof features is also greatly influenced by outliers.

The development of laser scanning technologies has permitted the gathering of large volumes of LiDAR point clouds at high frequencies and densities, allowing for the derivation of more detailed 3D models. In addition, the demands for robust data management, efficient spatial data retrieval and high reconstruction performance are also desirable. Although several reconstruction approaches have provided significant progress toward the automatic and efficient generation of 3D building models, this process is still tedious and time-consuming, especially for massive point clouds, and economical and reliable techniques that fully exploit the advantages of spatial databases are lacking.

1.1. *Objectives of this work*

This paper proposes a novel spatial database-based solution framework for reconstructing 3D building roofs with high quality and performance. To improve performance, the framework maximizes the parallel computation capabilities offered by spatial databases. To obtain robust results, outliers are removed during the reconstruction steps, and the primary roof structures are determined from large numbers of planar points, while the roof details are fitted with primitives. The framework works in a fully automatic manner by integrating data-driven and model-driven methods to model building roofs with primary structures and detailed features.

The general strategy for the building roof reconstruction framework implemented in this paper is given in Figure 1. The LiDAR data used in this framework have been classified as building regions (Zhao *et al.* 2016). The building roof modeling begins with individual building segmentation and approximate boundary determination. The individual building segmentation process is based on a density-based spatial clustering technique using a spatial index to improve performance (Section 2.1). After tracing the approximate boundaries (Section 2.2), Delaunay triangulation with boundary constraints is applied, and the normalized vector of each triangle is computed and quantified in parameter space. The histogram of normal vectors is accumulated to detect the local peaks, with each peak representing similar attributes (parallel or coplanar planes). Starting from a triangle with these attributes, a spatial connectivity analysis is performed in object space, iteratively growing to obtain an initial roof plane and thus separating coplanar or parallel planes (Section 2.3). Boundaries are regularized based on a least squares method (Section 2.4) and are adjusted according to the directions of the long ridges. Roof details are fitted with primitives by analyzing the remainder of the non-planar points with parallel, orthogonal, horizontal or vertical hypotheses. Lastly, topological and geometrical information of 3D roof models are derived from these primary planes and details (Section 2.5). Using a spatial database, multiple building roof models are simultaneously reconstructed.

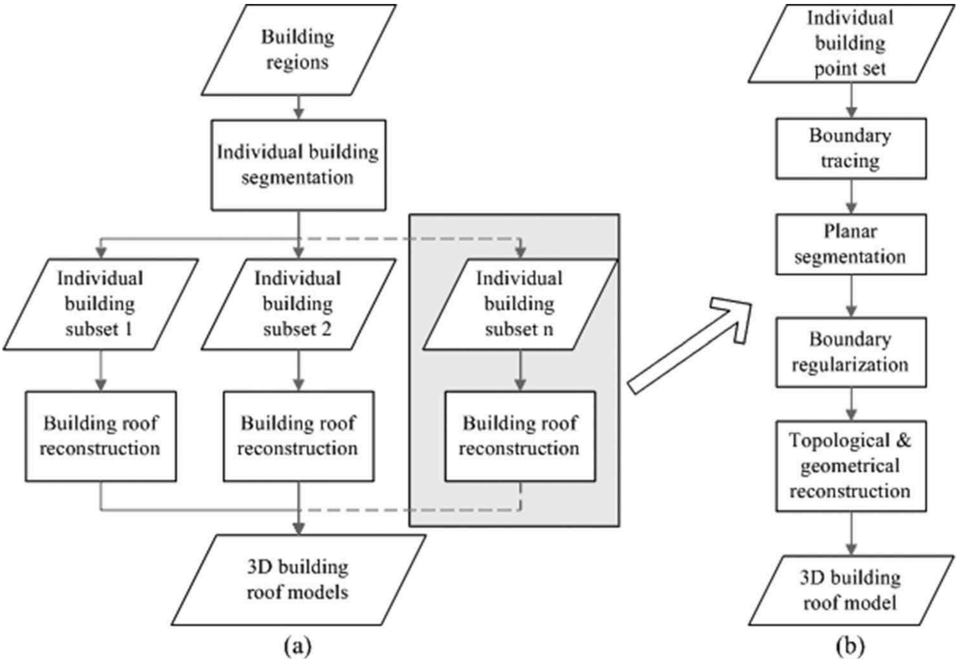
Building regions
Individual building segmentation
Individual building subset 1
Individual building subset 2
Individual building subset n
Building roof reconstruction
Building roof reconstruction
Building roof reconstruction
3D building roof models
(a)
Individual building point set
Boundary tracing
Planar segmentation
Boundary regularization
Topological & geometrical reconstruction
3D building roof model
(b)

Figure 1. Flowchart of 3D building roof reconstruction: (a) the framework and (b) the reconstruction process for an individual building.

1.2. *Related work*

The automatic reconstruction of 3D building models from airborne LiDAR point clouds has progressed significantly over the past few decades (Axelsson 1999, Haala and Kada 2010, Rottensteiner *et al*. 2014). In general, reconstruction approaches can be categorized as data driven or model driven. Data-driven reconstruction techniques (Verma *et al*. 2006, Sampath and Shan 2010) can produce more accurate and robust models and details from denser point clouds. Using predefined primitives, model-driven methods (Henn *et al*. 2013, Huang *et al*. 2013, Xiong *et al*. 2015) can flexibly reconstruct building models from sparser data but are still challenged by highly sophisticated shapes. A combination of the two approaches (Kwak and Habib 2014) can make improvements, such as by obtaining more data hints to combine various primitives or filling data gaps with model knowledge.

When classified from raw LiDAR data, a dataset of building regions usually contains numerous buildings. To produce 3D models, individual buildings need to be segmented, for which the commonly used segmentation techniques are based on clustering. A wide range of cluster models (e.g. connectivity, centroid and distribution models) and clustering algorithms (e.g. hierarchical clustering, *k*-means and expectation maximization) are commonly used (Estivill-Castro 2002). Sampath and Shan (2007) proposed a region-growing approach to group similar points into the same building by iteratively collecting points within a moving window (MW). Kwak and Habib (2014) and Lari and Habib (2014) derived individual building hypotheses with traced boundaries using the modified convex hull algorithm (Sampath and Shan 2007). However, the above methods are sensitive to noise and computationally expensive.

To trace building boundaries, convex-hull-like algorithms or modifications are commonly used, as well as α-shape-like algorithms (Edelsbrunner *et al*. 1983). The efficient computation of convex hulls has been a topic of much research (Graham 1972, Jarvis 1973, Eddy 1977, Preparata and Hong 1977, Andrew 1979, Barber *et al*. 1996). To extract concave outlines, Sampath and Shan (2007) presented a modified convex-hull method that limits the candidates in a local rectangular neighborhood, while Moreira and Santos (2007) restricted the neighborhood within the *k*-nearest neighbors. In contrast to the convex or concave methods, the α-shape-like algorithms (Edelsbrunner *et al*. 1983, Edelsbrunner and Mücke 1994, Melkemi and Djebali 2001, Cazals *et al*. 2005) can extract inner boundaries but are limited by the determination of a proper α value. Mandal and Murthy (1997) proposed techniques for estimating the best *α* to determine the boundary. Dorninger and Pfeifer (2008) used an *α* value of twice the average point space of the point clouds. However, a prerequisite for these approaches is very few outliers in the dataset.

When being traced, zigzagged boundaries need to be regularized to represent the real outlines. The Ramer–Douglas–Peucker algorithm (RDP) (Ramer 1972, Douglas and Peucker 1973) is commonly used to simplify bounding lines; however, it may erroneously discard critical boundary points (Zhang *et al*. 2006). To improve the results, constraints with parallel, orthogonal or dominant directions were introduced to adjust the building boundaries (Zhang *et al*. 2006, Sampath and Shan 2007, Dorninger and Pfeifer 2008, Kwak and Habib 2014).

As important primitives, planar patches are of particular interest for building reconstruction. Generally, they are segmented in spatial space or in parameter space based on similarity and proximity. Region growing with seed points is one of the most commonly utilized spatial domain methods to extract planar facets (Vosselman and Gorte 2004, Rabbani *et al*. 2006, Lari and Habib 2014) but is sensitive to the selection of seed points and not robust enough (Wang and Shan 2009). Random sample consensus (Fischler and Bolles 1981) is a relatively robust but time-consuming approach used to estimate planar parameters (Verma *et al*. 2006). In contrast to spatial domain segmentation approaches, the selection of seed points is not necessary for methods that segment planes in the parameter domain, yet the results depend heavily on the estimated attributes that are usually derived within a local neighborhood. The most common attributes of a plane are the normal vector and the position vector, which are computed for a particular point from different neighborhoods (Axelsson 1999, Vosselman and Gorte 2004, Filin and Pfeifer 2006, Lari and Habib 2014, Kim *et al*. 2016). As a result, the classical Hough transformation (Duda and Hart 1972, Ballard 1981) has been extended to detect planes (Vosselman and Dijkman 2001). However, parameter domain segmentation approaches are still inefficient, especially for large numbers of points (Lari and Habib 2014).

Boundary representation (B-rep) is widely used to represent 3D building models (Stroud 2006). There are two types of information in a B-rep, topological and geometric. The topological and geometrical items, such as faces (surfaces), edges (curves) and vertices (points), can be determined by the intersection of planar patches. With the help of the introduced auxiliary vertical planes, inner and outer roof corners are derived from the intersection of the adjacent roof planes (Sampath and Shan 2010, Xiong *et al*. 2014). However, the determination of step edges is still challenging as well as the reconstruction of detailed roof features (Haala and Kada 2010, Rottensteiner *et al*. 2014).

GIS or spatial databases are commonly used to robustly and efficiently manage massive spatial datasets. van Oosterom *et al.* (2015) conducted some benchmark tests on various spatial databases to evaluate their performance in loading and retrieving point clouds, which may help establish performance improvements. For example, a full exploitation of multi-process architectures may provide faster data retrieval when reconstructing building models from large quantities of raw LiDAR data. However, there have been no comprehensive tests of the performance of 3D building reconstruction from very large datasets.

2. Methodology

In this section, the spatial database-based framework for building roof reconstruction from airborne LiDAR points is addressed. To trace building boundaries, the framework starts by partitioning the building regions dataset into individual building subsets using a density-based clustering approach. Then, a hybrid method that selects seed points in the parameter domain and grows the regions in the spatial domain is used to extract the planar patches. Boundary regularization is applied after removing noise and 3D models are obtained from faces, edges and vertices which are determined by the intersection of planar patches.

2.1. *Individual building segmentation*

The goal of individual building segmentation is to cluster building points into different groups, whereby each group represents only one building and each point belongs to only one building. In this paper, the density-based spatial clustering of applications with noise (DBSCAN) (Ester *et al.* 1996) is applied to segment the building point sets into individual buildings with arbitrary shapes (Estivill-Castro 2002). As a density-based clustering method, DBSCAN characterizes a well-defined 'density-reachability' cluster model by connecting the points that satisfy a density criterion defined as a minimum number of objects within certain distance thresholds. The algorithm is implemented as

(1) Begin with an arbitrary unvisited point P_0 that has at least *minPts* adjacent points within its ε-neighborhood, otherwise, the point is labeled as noise.
(2) Add P_0's ε-neighborhood points, if their ε-neighborhoods are also dense.
(3) Repeat until the density-connected cluster is completely found.
(4) Retrieve and process a new unvisited point as above.

In the above procedure, the spatial index based on the Generalized Search Tree (Hellerstein *et al.* 1995) for point datasets offers improvements on ε-neighborhood retrieval, and the spatial database provides capabilities for parallel computation of clustering points. After the above operations, all isolated noise points are removed, and all building points are assigned to specific buildings. Figure 2 shows an example of the segmentation of individual buildings.

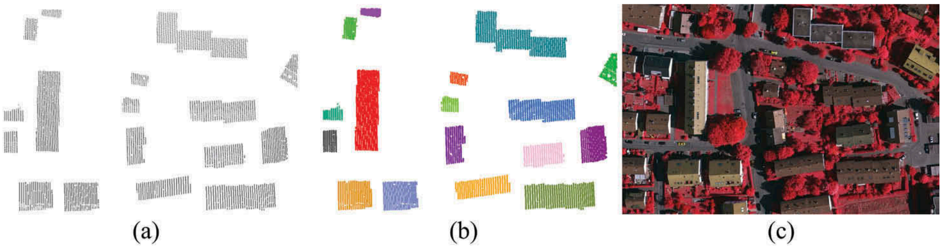

Figure 2. Individual building segmentation: (a) input building regions, (b) segmentation results and (c) reference image.

2.2. *Boundary tracing*

There are two major categories of methods used to trace building boundaries, raster based and vector based. In the raster-based methods, the point clouds are converted to regular grid data, and image processing techniques are applied to identify, trace and regularize the boundary edges (for more details, see Zhang *et al*. 2006, Zhou and Neumann 2008, Awrangjeb *et al*. 2010, Grigillo and Kanjir 2012). In the vector-based approaches, a simple, closed-line string (Linear-Ring) defines the exterior boundary, and zero or more interior Linear-Rings (OGC 2006) representing the inner boundaries are extracted from the input point clouds.

In this paper, the modified convex hull boundary detection algorithm (Sampath and Shan 2007) is redesigned and refined using a spatial database to determine the boundaries of the individual buildings. The first modification is the introduction of a minimum number of neighbors (*minPts*) for a border point. If the number of points in the rectangular neighborhood is less than the threshold, the *k*-nearest neighbors are supplied. The reason for this modification is that the determination of an appropriate distance threshold is difficult (especially for a nonuniformly distributed point set), and insufficient neighbors may result in failure when using the modified convex hull approach. The second modification is that the point spacing in along and across scan directions is replaced by the average point spacing because the latter is more easily computed from the average point density. The boundary tracing (BT) process is

(1) Find the start boundary point P_0 that has the lowest y-coordinate. If more than one point has the lowest y-coordinate, the point with the largest x-coordinate is chosen.
(2) Find the next boundary point P_1 from the neighborhood point set of P_0 that has at least *minPts* points where the rotation angle from the sweep line vector $(0, -1)^T$ or negative y-axis to the candidate point is the minimum in the counter-clockwise direction. If collinear points exist, the point with the shortest distance to P_0 is selected.
(3) Repeat step (2) until point P_{k+1} is equal to P_0, where the sweep line is replaced with the vector $\boldsymbol{P}_k\boldsymbol{P}_{k-1}$ within the neighborhood of P_k.

Figure 3 illustrates the above procedure. The algorithm selects the lowest point [red dot in Figure 3(a)] as the start point. The rotation angle from the initial ray, vector (0,

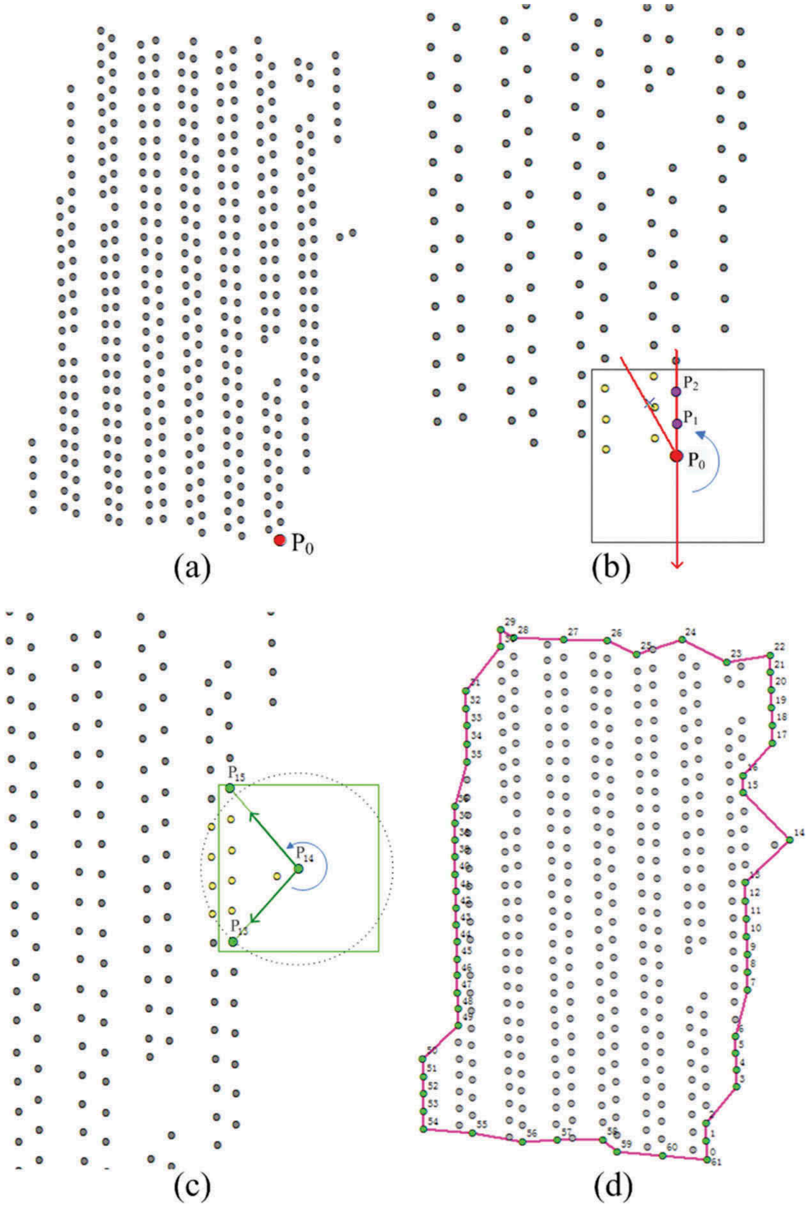

Figure 3. Building boundary tracing: (a) the red dot has the lowest *y*-coordinate that is selected as the start point P_0; (b) using convex hull algorithm to find the next boundary point in P_0's neighbors (yellow and purple points inside the box), and P_1 with a shorter distance to P_0 is selected from collinear points (purple colored); (c) the neighbors of P_{14} within the box neighborhood are insufficient and the *k*-nearest neighbors (yellow dots outside the green box) are supplied; and (d) traced boundary (green vertices ordered in the counter-clockwise direction).

$-1)^T$ [red line in Figure 3(b)], to each adjacent point of P_0 (counter-clockwise direction) is computed. As shown in Figure 3(b), the purple points have the same minimum rotation angle (collinear points), but only the point with a shorter distance to P_0 is selected as the next boundary vertex [P_1 in Figure 3(b)]. Figure 3(c) presents an example of insufficient points within the rectangular neighborhood of P_{14}. In this case, the circular adjacent points sorted by their distance to P_{14} (yellow dots outside the green box) are supplied with the rectangular neighbor points (yellow dots within the green box). Figure 3(d) shows the traced building boundary (purple lines sequentially connecting the green points).

2.3. *Roof planar segments*

High-quality extraction of planar surfaces is a critical issue in reconstructing 3D building models since planar surfaces serve as important primitives. Generally, there are two categories of techniques for segmenting planar features from LiDAR data, that is, segmentation in spatial (object) space and segmentation in parameter (attribute) space.

A hybrid spatial database-based approach is introduced in this paper to segment planar patches on rooftops. In the first step of the approach, after tracing approximate boundaries, boundary constrained Delaunay triangulation (Shewchuk 2002) is applied to the individual building point set, and the normal vector of each triangle is computed. Then, the unit normal vector $\boldsymbol{v}$ of each triangle is quantified in attribute space, where each component of $\boldsymbol{v}$ is within the range of $[-1, 1]$. The histogram of the unit vectors is accumulated to detect peaks, and every local peak represents similar attributes of planar normals (parallel or coplanar planes). Starting from a triangle with these attributes, a spatial connectivity analysis is performed in the object space, iteratively growing to obtain an initial plane, thereby separating coplanar or parallel planes. Then, the same grouped triangles are merged to form a polygon in 3D space, and the parameters of the least squares plane are estimated by analyzing the vertices associated with these triangles. The method presented in this paper offers the following advantages: (1) the seed points selected are robust as they are local peaks accumulated in parameter space, and (2) region growing in the spatial domain is limited to the subset with similar attributes, not the universal set.

2.4. *Boundary regularization*

Generally, a set of LiDAR points next to building border areas is susceptible to containing outliers (vegetation, power line or wall points). After some of the outliers are removed during the density-based clustering process (Section 2.1), the process continues by analyzing the residuals of the boundary candidates to their nearest plane. If the residual is larger than a given threshold, it is excluded as noise. Therefore, boundaries are regularized after the planar patch extraction process.

The boundary regularization method is based on a least squares technique, and the processing flow is illustrated in Figure 4. Before fitting a line segment from a subset of homogeneous boundary points in the vicinity, the points are grouped by calculating the slope angle of each line segment with the end points of two immediate adjacent boundary vertices and iteratively verified within two consecutive ones. If the two neighbors of a short line segment are almost parallel, they are merged into one group and the line parameters are recomputed using the least squares method. Otherwise, if the two adjacencies are nearly perpendicular, they are elongated to their intersection point, and the end points of the short line are merged. Due to the nature of laser

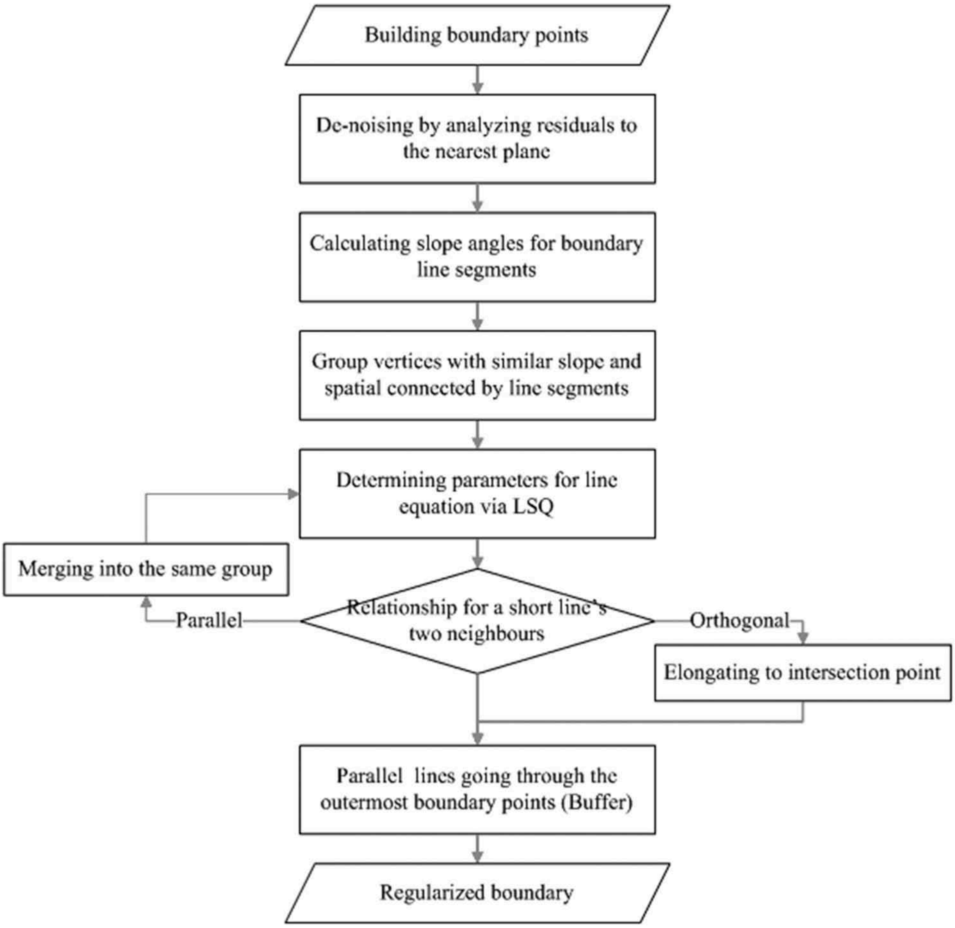
Building boundary points
De-noising by analyzing residuals to the nearest plane
Calculating slope angles for boundary line segments
Group vertices with similar slope and spatial connected by line segments
Determining parameters for line equation via LSQ
Merging into the same group
Parallel
Relationship for a short line's two neighbours
Orthogonal
Elongating to intersection point
Parallel lines going through the outermost boundary points (Buffer)
Regularized boundary

Figure 4. Flowchart of the boundary regularization.

scanner data acquisition, buffers are applied to every border line segment, that is, a parallel line going through the outermost boundary vertex.

2.5. *Building roof reconstruction*

A spatial model of a 3D building roof is described in the form of B-rep in this paper. B-rep models are composed of two parts (Stroud 2006), geometry (surfaces, curves and points) and topology (the main items include faces, edges and vertices). Edges and vertices can be determined by their intersecting adjacent planes. After the planar patches are derived, roof models can be reconstructed by intersecting these segments to determine break lines and vertices. For each pair of adjacent planar segments, an intersection line can be deduced. For every boundary line segment, a vertical plane is assumed to intersect with a corresponding roof planar segment. All roof vertices (interior and exterior points) are determined by the intersection of the adjacent roof planes, including the assumed vertical planes through the boundary.

Typically, step edges exist in the adjacent regions between pairs of neighboring parallel planar segments. When planar patches are clustered within connected triangles with homogeneity in a Delaunay triangulation of airborne LiDAR points, step edges can barely be detected, as the triangles constructed from the points near the step-edge area tend to form an inclined plane; thus, a pseudo-ridge is intersected. The step edges are recovered by assuming that the planar roof polygon, which connects two parallel planes, has a particular area and shape factor, that is, a small area and a narrow shape. If so, a vertical plane through the midline of the polygon (parallel to the adjacent edges of the parallel planes) is introduced, and the intersection lines lying between the corresponding parallel planes are determined. Additionally, the parallel planar patches are extended to their respective intersection lines.

When the primary structure of a building roof has been constructed from planar patches and their intersection lines and vertices, some detailed features may still exist on the roof that are difficult to model directly due to sparse LiDAR points. Hence, model primitives are applied to reconstruct these details. They are obtained by analyzing the remaining nonplanar points, and parallel, orthogonal, rectangular, horizontal or vertical hypotheses are used to fit them. If a horizontal plane can be fitted with a set of connected nonplanar points near the intersection region of a pair or more of planar segments, a minimum horizontal rectangle containing all fitted points is regularized with the property that one of its sides is parallel to the nearest long ridge. Correspondingly, vertical planes containing the rectangle's edges are inserted to create the water-tight part of the roof, the intersection lines and vertices are calculated and the planar roof patches are updated. Similarly, if a nonplanar point set is bounded within a single planar segment and the fitted plane is parallel to the segment, one edge of the adjusted rectangle is parallelized to the nearest long edge of the segment; but whether vertical or orthogonal planes are inserted depends on the distance between the parallel pair. Then, a 3D building roof model is reconstructed from these patches, lines and vertices, with which every triangle face of the model is derived in a triangulation with the constraints of the edges. To reconstruct numerous roof models efficiently, the spatial database is used to process multiple individual buildings simultaneously. Additionally, the reconstruction of 3D building roof models is fully automatic.

3. Experiments

The Vaihingen dataset (Cramer 2010) was selected to fully test the new building roof reconstruction method. The airborne LiDAR point cloud was acquired using a Leica ALS50 laser scanning system. The accuracy is 10 cm in planarity and vertical and the average point density varies from 4 to 6.7 points/m^2 (Rottensteiner *et al*. 2014). The selected test site is characterized by detached residential buildings with various structures (Figure 2), and the input point cloud has been classified as building points (Zhao *et al*. 2016). The framework is implemented in PostGIS, which extends PostgreSQL with robust spatial database management capabilities and offers various spatial analysis functions.

To evaluate the effectiveness of the proposed individual building segmentation method, we compared it with the MW algorithm (Sampath and Shan 2007) and the new BT method (Awrangjeb 2016). In the following sample dataset (Figure 5(a)), there

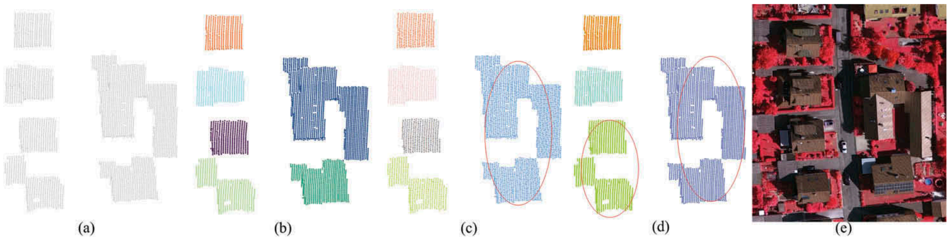

Figure 5. Individual building segmentation: (a) input point cloud (building regions), (b) segmentation results of the proposed method w.r.t. ε = 1.0, *minPts* = 5, 2D Euclidean distance (colored differently), (c) segmentation results of the moving window method w.r.t. ε = 1.0, 2D Euclidean distance (the red circled two buildings are erroneously grouped in one cluster), (d) segmentation results of the new boundary tracing method (the two buildings on bottom-left and the two buildings on the right are erroneously grouped in same clusters) and (e) reference image.

are six gable-roofed houses with differing shapes and the two buildings on the right are very close to each other (the distance between the two nearest points is 0.85 m in 2D and 0.86 m in 3D). All three methods used the 2D Euclidean distance. As seen in Figure 5 (b), the proposed method correctly segmented all six buildings (colored differently), while the MW method could not differentiate the right two buildings (Figure 5(c), circled in red), and the BT method grouped the right two buildings and the bottom-left two buildings into the same clusters [Figure 5(d), circled in red].

Figure 6 presents an example of the segmentation of planar roof patches. In principle, the triangulation of a planar point set results in a convex hull. The convex border does not always fit well with the real building boundary. Correspondingly, some of the derived triangles outgrow the building regions (Figure 6(d)). To improve the quality, the point set was triangulated with the constraints of the traced boundaries (Figure 6(c)). Due to the accuracy of data acquisition, quantization errors or computation errors, coplanar triangles may have differences in their normal attributes and thus result in holes or gaps when clustering coplanar triangles using the region growing approach (Figure 6(h)). It is known that if a set of points lie in the same plane, the triangles triangulated from them must be coplanar and vice versa. Therefore, when a triangle has dissimilar attributes with its two or three 'side' adjacent triangles that have the same properties, its attributes are updated. In Figure 6(h), the blue circled triangles (marked as A) differ from their three neighbors, while the dashed cyan circled triangles (marked as B) differ from the twos, but are all segmented in the same group within their vicinities. After the clustered triangles were merged into planar polygons (Figure 6(g)), plane parameters were computed using a least squares method, using a form of '$z = Ax + By + D$' (for building roof planar patches derived from airborne LiDAR data, the v_z component of the normal vector is not equal to zero).

The results of boundary regularization illustrate that the proposed method is capable of deriving high-quality building boundaries. Figure 7 illustrates some implementation details of the boundary regularization method. Boundary points are grouped by their spatial connectivity and the homogeneity of the slope angles, thereby avoiding the drawbacks of least squares methods (susceptible to outliers) and the drawbacks of the RDP (missing critical corners). For the two neighbors of a short boundary line segment,

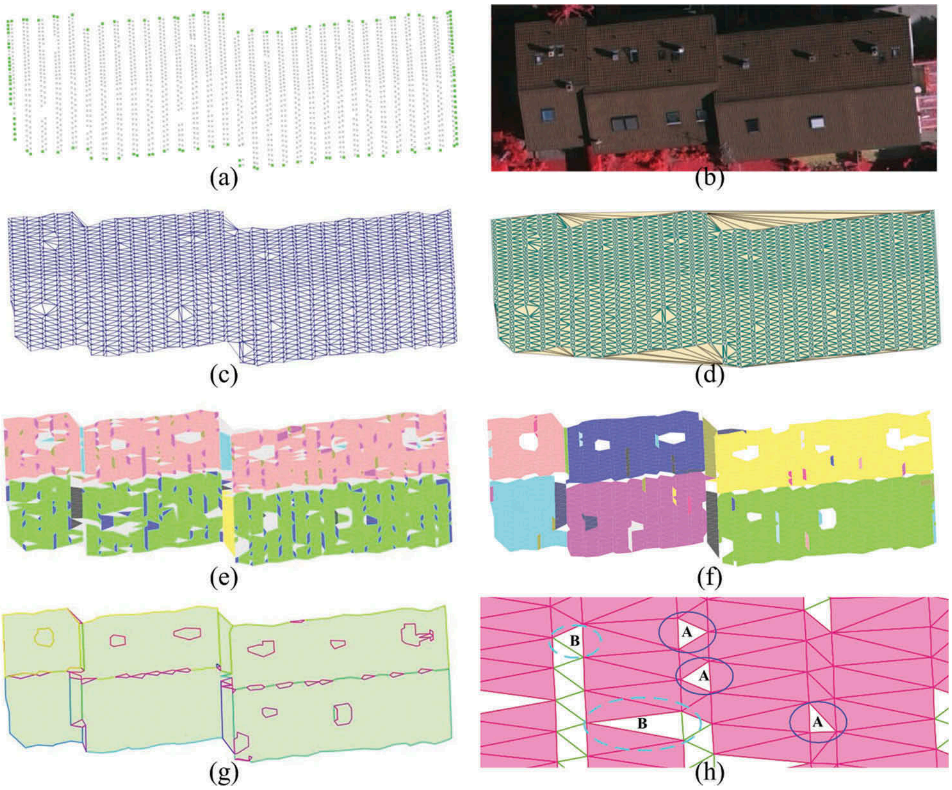

Figure 6. Planar segmentation: (a) an individual building point set with traced boundary (green colored), (b) reference image, (c) Delaunay triangulation with boundary constraint, (d) Delaunay triangulation without constraint, (e) triangles grouped by similar normal vectors, (f) planar patches clustered in the spatial domain, (g) planar polygons derived from planar segments and (h) coplanar triangles not clustered due to quantization errors or computation errors.

the relative location relationship is analyzed. If the segment tends to be orthogonal or parallel, the line segments are intersected or refitted, respectively (Figure 7(c)). The fitted line equation is a form of Equation (1) or Equation (2), depending on the line segment's tendency to be horizontal or vertical.

$$y - \bar{y} = A(x - \bar{x}), \tag{1}$$

$$x - \bar{x} = A(y - \bar{y}), \tag{2}$$

where $\bar{x}$ and $\bar{y}$ denote the average of $\sum x_i$ and $\sum y_i$, respectively.

The linear equations fitted in the horizontal plane are applied to vertical planar equations in 3D to determine the intersection lines (Section 2.5). In general, the least squares results indicate minimized fitting errors, but when fitting building borders, it will exclude some roof parts (Figure 7(d)). To solve this problem, buffered lines going through the outmost boundary points are used to improve the least squares results (Figure 7(e,f)).

The reconstructed roof models are composed of geometrical and topological information (faces, edges and vertices). After some nonplanar point sets within a planar

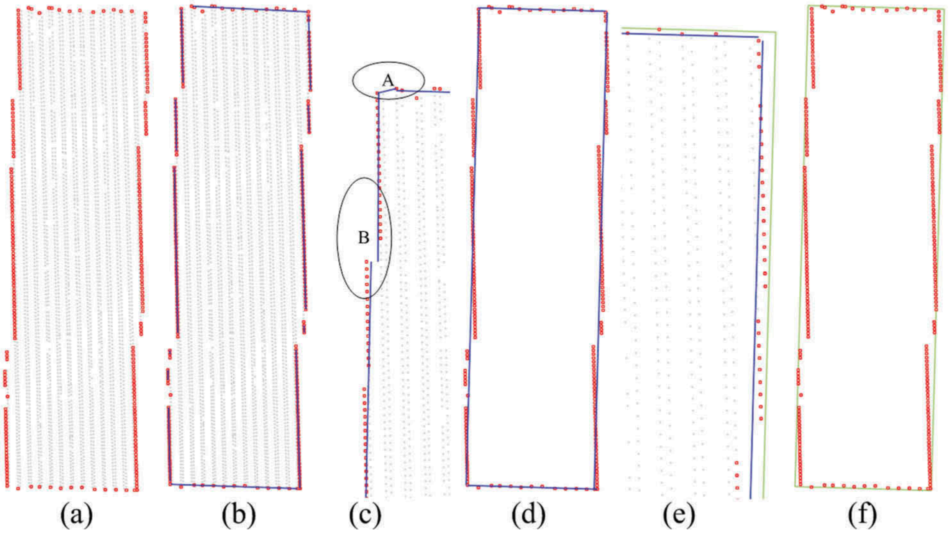

Figure 7. Boundary regularization: (a) point cloud with traced boundary vertices, (b) least squares lines fitted with the point sets with similar slope angle and spatial proximity, (c) nearly orthogonal pairs of line segments intersecting (circled with A) and parallel pairs refitting (circled with B), (d) fitting results, (e) buffered lines (green colored) through the outmost points and (f) regularization results.

segment are discarded due to counts that are insufficient to support a robust plane (at least four noncollinear points within a connected region) and the corresponding isolated triangles are merged, reconstruction starts with the intersection of the pairs of adjacent planes (Figure 8(a)). Roof break lines and corners are determined by the intersection of planar patches (Figure 8(b)), and auxiliary vertical planes are introduced to locate the step edges and boundaries. As some triangles are not grouped into planar segments, which mainly lie near intersection lines, an area threshold of 1 m^2 is used to filter small polygons to avoid intersecting them with others. In general, intersection lines of pairs of planar patches are not coincident with their boundaries, and it is necessary to reconstruct the faces with the intersection lines and vertices, that is, the planar patches are extended to or clipped by the intersection lines.

Detection and determination of step edges are still challenging. When a narrow and small polygon is adjacent to two parallel planar patches, there is almost a step edge. In this case, the two intersection lines are almost parallel [Figure 8(b), purple-circled groups], and a vertical plane through the midline of the parallel lines that intersects with each of the particular planar patches is used to determine the step edge (Figure 8 (d)). Furthermore, another type of pseudo-ridge is incorrectly deduced from two adjacent planar segments [Figure 8(b), red line segment], that is, the planes intersect, but they share few common parts, and when crossed by planar boundaries, the most part lies within the respective patch, and thus, this line segment is clipped (Figure 8(c)). Once a ridge line is determined, it serves as a reference to orthogonalize or parallelize the other sides of the patch if it satisfies an angle condition. Again, intersection points are determined by extending or intersecting two consecutive line segments, and the planar roof patches are updated with these vertices (Figure 8(e)).

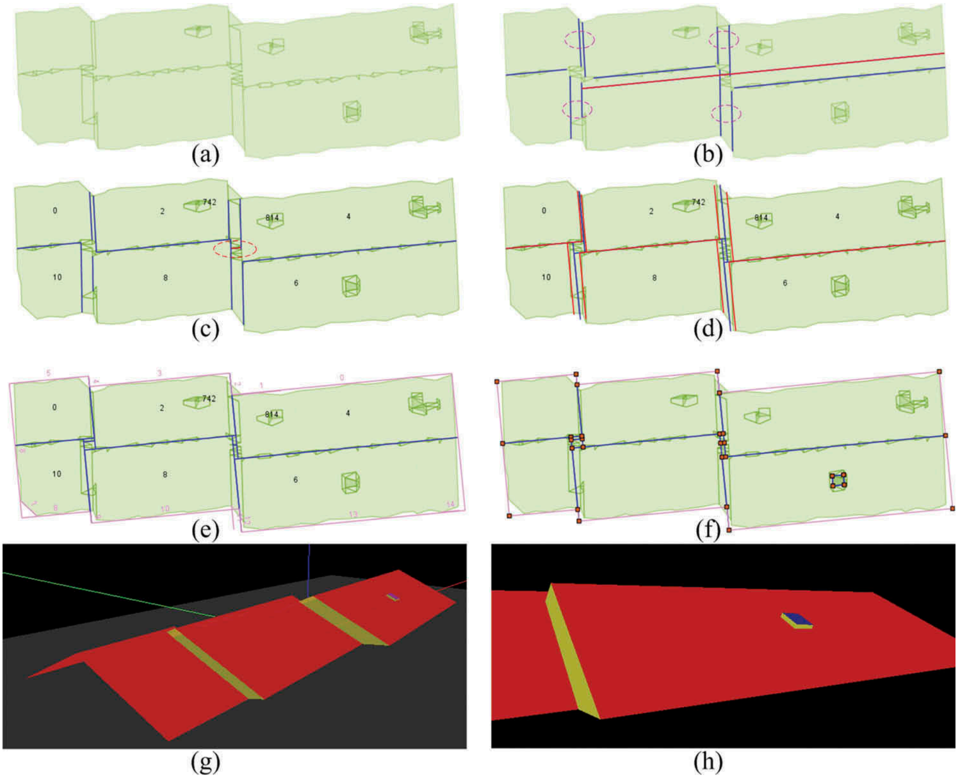

Figure 8. Building roof reconstruction with step edges: (a) planar patch extraction results, (b) intersection lines of adjacent pair of planes, (c) result of pseudo-ridge clipping, (d) determination of step edges, (e) boundary regularization (adjusting the other sides of a planar polygon referring to the long intersection line), (f) fitting details, (g) 3D roof model and (h) detailed roof feature in 3D perspective view.

To recover the roof details, parallel, orthogonal, horizontal, vertical or rectangular hypotheses are used to analyze the remaining nonplanar points. If can fit a parallel plane with a point set within a particular planar segment, a minimum rectangle containing these points is adjusted that one side is parallel to the nearest long edge of the segment (Figure 8(f)). The vertical or orthogonal planes through the sides of the rectangle are inserted depending on the distance of the parallel pair to create the water-tight part of the roof (Figure 8(h)). Similarly, when fitting a horizontal plane near the intersection regions of a pair or more of planar patches, one edge of the rectangle is regularized parallel to the nearest long ridge (Figure 9). Then, a 3D building roof model containing faces, edges and vertices is reconstructed from the primary structure and details (Figure 8(g)).

Figure 9 shows an example where detailed features satisfy a horizontal plane condition, and the rectangles are regularized to align with the nearest long intersection line of planar segments. Note that these details are enclosed with vertical planes intersecting particular roof segments [cf. Figure 8(h), auxiliary planes orthogonal to the planar patch], and the nonplanar points that cannot robustly fit a plane are discarded (Figure 9(b)). Figure 10 is an illustration of the reconstruction of a flat roof with parapet walls

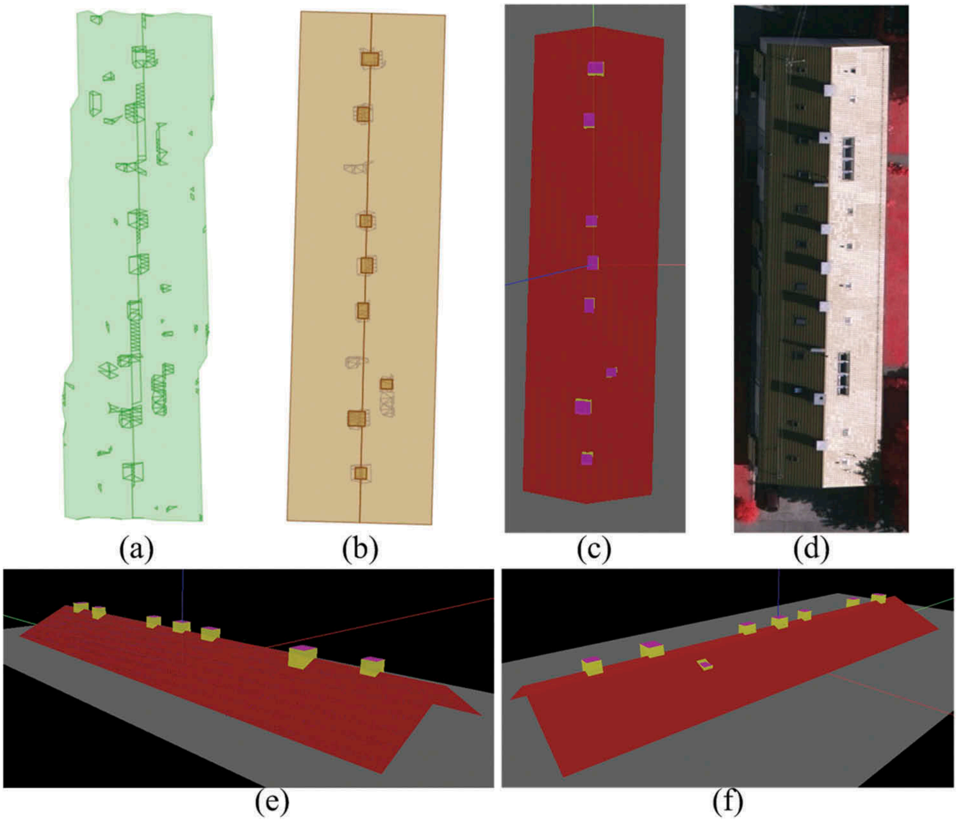

Figure 9. Gable roof reconstruction with details: (a) planar segmentation results, (b) reconstruction result in 2D view, (c) reconstruction result in 3D view, (d) reference image, (e) 3D perspective view (with raw LiDAR points) and (f) detailed 3D roof model (in perspective).

(Figure 10(d)). As can be seen, dozens of nonplanar points near the border (Figure 10(b)) are removed, and the corresponding regions are merged into the primary planar patch since they cannot fit robust planes. Boundary noises are removed by the fitting residual on the nearest plane, which has a threshold of 0.75 m (Figure 10(a)). Figure 11 shows other examples of building roof reconstruction.

We also conducted experiments on the performance evaluation for parallel computation based on the spatial database (Table 1). Although more comprehensive and detailed tests should be conducted, the preliminary results show that the time spent on the reconstruction of multiple building roofs using parallel computation is similar to that spent on the reconstruction of a single roof but far less than the time spent in serial mode. Furthermore, the segmentation of individual buildings can be parallelized by partitioning the dataset into subsets. As a result, the robust spatial data management capabilities and effective spatial analysis functions offered by the spatial database greatly improved the automatic reconstruction of 3D building models, especially given the increasingly large volumes of data acquired by laser scanning systems due to their high frequencies and increasing point densities.

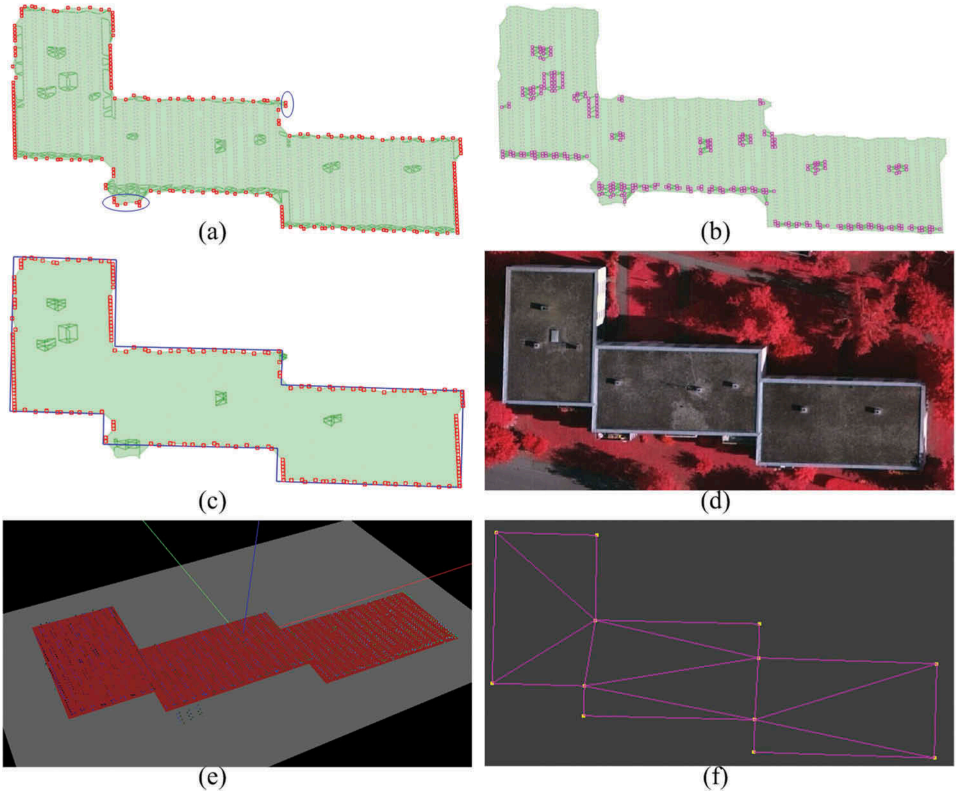

Figure 10. Reconstruction for a flat roof: (a) planar segmentation (gray and red dots are inner and boundary points, respectively), (b) nonplanar points (purple colored), (c) de-noise and boundary regularization, (d) reference image, (e) 3D roof model (with raw points) and (f) geometrical faces, edges and vertices.

4. Discussion

The settings of the thresholds, such as the distance and the point count, and some implementation details of the aforementioned steps are presented in this section.

The first issue is the calculation of the average point spacing of airborne LiDAR point clouds. The flight height, flying pattern, scanning pattern, platform movements, strip overlap and object properties all contribute to the variations in the point distribution in along or across-track directions and result in a nonuniformly distributed point set. The average point spacing, d, derived from the mean point density, m (points/m^2), was adopted in this paper, namely, $d = m^{-1/2}$, where m is given by the dataset or calculated by the total number of points divided by the area.

When separating individual building point sets using the density-based spatial clustering algorithm DBSCAN (Ester *et al.* 1996), the distance ε and the minimum number of points *minPts* are required. Whether a point is a valid candidate depends on if its ε-neighborhood contains a sufficient number of *minPts* points. The smaller ε, the fewer points in the dataset will be clustered, whereas for a too high value of ε, clusters tend to merge the majority of the points into the same cluster. In general, a small value of ε is

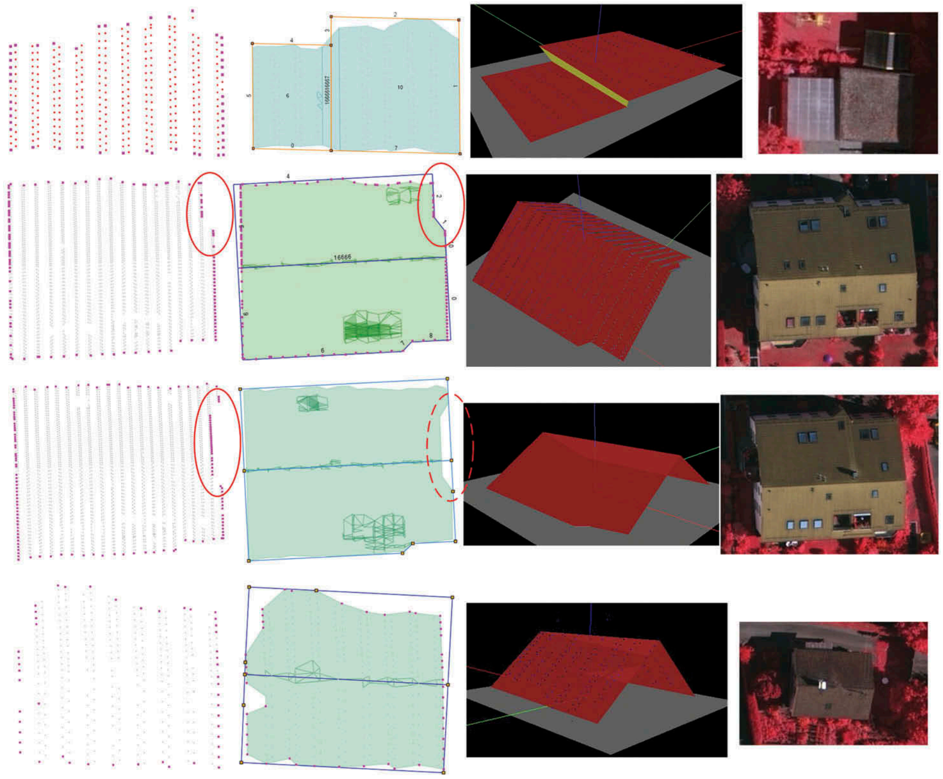

Figure 11. Examples of roof reconstruction (from left to right): individual building point sets (with traced boundary points in purple color), geometrical elements (in 2D), roof models (in 3D) and reference images. Some missing parts of the buildings (circled in red) recovered successfully (dashed red circled).

Table 1. Performance evaluation for building roof reconstruction.

Building number	Number of points	Number of triangles	Runtime (s) (serial mode)	Runtime (s) (parallel mode 1)	Runtime (s) (parallel mode 2)
1	346	634	4.961	16.864	23.695
2	2228	4336	54.117	86.440	141.556
3	394	727	4.539	14.368	22.307
4	3290	6322	74.475	105.394	144.550
5	2459	4770	55.740	90.418	141.510
6	340	625	3.915	13.774	18.297
7	1100	2064	15.694	29.219	43.071
8	266	488	4.352	12.604	13.620
9	1780	3437	26.067	51.184	106.565
10	1905	3674	26.739	65.519	125.253
11	2616	5018	45.069	69.265	133.505
12	1304	2490	17.440	41.776	53.399
13	3438	6710	86.657	105.551	169.401
14	1737	3346	30.202	67.362	128.794
Total	23,203	44,641	450.609	211.304	169.557

(1) All tests run on the same platform, a Lenovo® T4900d-00 PC with 4 GB RAM, Intel® Core(TM)-i7-3770 CPU @3.4 GHz, ATA ST1000DM003 disk, Windows7 64 bit, PostgreSQL 9.4 and PostGIS 2.2; (2) parallel mode 1 is 8-channel and mode 2 is 16-channel.

preferable, and a value of twice the average point spacing is selected in this paper. As a rule of thumb, *minPts* can be derived from the dimensions D of the dataset, that is, $minPts \geq D + 1$. A larger value of *minPts* will yield a better result for a dataset with noise. Considering that the point clouds have been filtered and contain only building regions, we use a value of 4 for *minPts*. When implemented in a spatial database, the spatial index can provide ε-neighborhood retrieval with an $O(\log n)$ time complexity, and an overall average runtime complexity of $\mathrm{O}(n \log n)$ can be expected. This is a better result, but the time to process massive LiDAR point clouds it still very long. In practice, the spatial database can be feasibly used to partition the dataset into subsets for parallel processing. While the runtime complexity remains unchanged, the processing time is reduced. This work is planned for the near future.

The third issue is related to the parameter settings, the neighborhood distance ε and the neighbor count *minPts*, used in the BT process. The larger ε, the more points within a neighborhood will be swept, and the traced boundary tends to be more 'convex', as does the value of *minPts*. To retain as many boundary vertices as possible to fit a robust line segment, a small value of ε is preferred. However, an inappropriate value of ε may result in a failure (converging to a local loop, especially when points are not uniformly distributed) due to an insufficient number of neighbor points. Therefore, we introduced k-nearest neighbor points to fix this error (Section 2.2). Considering the nature of airborne LiDAR point clouds, the value of ε is set to twice the average point distance, and *minPts* is set as 9 to obtain robust results.

When segmenting planar patches, we select seed regions by detecting extremes in the histogram accumulated with triangle normals. In contrast to the normal of a point, the normal of a triangle is more intuitive and easily obtained. The parameter space of the unit vector $(v_x, v_y, v_z)^T$ forms a cube, and each component has a continuous range of $[-1, 1]$. If the v_z component is limited within $[0, 1]$, the space shrinks to half. To construct the histogram of unit vectors, we quantify the continuous space by an interval of 0.05, namely, the histogram has 40, 40 and 20 bins for v_x, v_y and v_z components, respectively. When converted to angles, the distribution of the bins of the histogram is a nonuniform quantization, but it needs no further calculation and satisfies the selection of seed points for planar growth. Additionally, we define 'adjacent' as two triangles with at least one common vertex.

5. Unsolved problems

The first unsolved problem is the determination of step edges. In this paper, we assume that a step edge exists near a pair of adjacent parallel planar patches, but this is not always true as step edges may also reside between nonparallel but discontinuous pairs. Additionally, the accuracy of the assumption that a step edge goes through the midline (or the midpoint) needs to be improved. A similar problem is the reconstruction of detailed roof features. The hypothesized primitives need to be verified and improved [Figure 9(f), an erroneously fitted dormer]. A dataset with higher point density or the incorporation of images or other data can improve the quality. Another problem is the polyhedral hypothesis, which cannot recover nonplanar surfaces. Although nonplanar surfaces can be approximated from a set of planar patches, the point density is a limitation, which is also work planned for the future.

6. Conclusions

This paper proposed a novel spatial database-based framework for the reconstruction of 3D building roof models from airborne LiDAR point datasets. The framework contains five major components: (1) a density-based clustering algorithm for segmenting individual buildings and detecting noise, (2) an improved BT algorithm that is robust to outliers, (3) a robust planar extraction method that selects seed points in the parameter domain and grows the regions in the spatial domain, (4) boundary regularization based on a least squares fit and (5) a method of intersecting planar patches and fitting primitives to reconstruct topological and geometrical information of building roofs with primary structures and detailed features. The entire reconstruction process is fully automatic and is implemented in a spatial database, which offers the advantages of efficiently managing and retrieving large volumes of spatial data and obtaining high performance via parallel computation.

Experimental results show that the proposed methods can properly and effectively model the detailed features of simple or complex polyhedral building roofs, as well as the primary structures. However, the reconstruction results for step edges and details were restricted by the density of the raw LiDAR points, which has been an ongoing challenge. Future work could improve the results by incorporating images and planning maps, as well as utilizing a denser point set.

Acknowledgments

The Vaihingen dataset was provided by the German Society for Photogrammetry, Remote Sensing and Geoinformation (DGPF) (Cramer 2010): http://www.ifp.uni-stuttgart.de/dgpf/DKEP-Allg.html (in German). Heartfelt thanks are also given for the comments and contributions of anonymous reviewers and members of the editorial team.

Disclosure statement

No potential conflict of interest was reported by the authors.

Funding

This work was supported by the National Natural Science Foundation of China under grant numbers 41571434 and 41322010.

ORCID

Xinyi Liu http://orcid.org/0000-0001-5333-8054
Zongze Zhao http://orcid.org/0000-0003-4869-9554

References

Andrew, A.M., 1979. Another efficient algorithm for convex hulls in two dimensions. *Information Processing Letters*, 9 (5), 216–219. doi:10.1016/0020-0190(79)90072-3

Awrangjeb, M., 2016. Using point cloud data to identify, trace, and regularize the outlines of buildings. *International Journal of Remote Sensing*, 37 (3), 551–579. doi:10.1080/01431161.2015.1131868

Awrangjeb, M., Ravanbakhsh, M., and Fraser, C.S., 2010. Automatic detection of residential buildings using LIDAR data and multispectral imagery. *ISPRS Journal of Photogrammetry and Remote Sensing*, 65 (5), 457–467. doi:10.1016/j.isprsjprs.2010.06.001

Axelsson, P., 1999. Processing of laser scanner data—algorithms and applications. *ISPRS Journal of Photogrammetry and Remote Sensing*, 54 (2–3), 138–147. doi:10.1016/S0924-2716(99)00008-8

Ballard, D.H., 1981. Generalizing the Hough transform to detect arbitrary shapes. *Pattern Recognition*, 13 (2), 111–122. doi:10.1016/0031-3203(81)90009-1

Barber, C.B., Dobkin, D.P., and Huhdanpaa, H., 1996. The quickhull algorithm for convex hulls. *ACM Transactions on Mathematical Software*, 22 (4), 469–483. doi:10.1145/235815.235821

Cazals, F., *et al*., 2005. Conformal alpha shapes. *In*: M. Alexa, *et al*., eds. *Symposium on point-based graphics (2005)*, 21-22 June 2005 Stony Brook, NY, USA. Switzerland: The Eurographics Association, 55–61. doi:10.2312/SPBG/SPBG05/055-061

Cramer, M., 2010. The DGPF test on digital aerial camera evaluation – overview and test design. *Photogrammetrie – Fernerkundung – Geoinformation*, 2010 (2), 73–82. doi:10.1127/1432-8364/2010/0041

Dorninger, P. and Pfeifer, N., 2008. A comprehensive automated 3D approach for building extraction, reconstruction, and regularization from airborne laser scanning point clouds. *Sensors*, 8 (11), 7323–7343. doi:10.3390/s8117323

Douglas, D. and Peucker, T., 1973. Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. *Cartographica*: *The International Journal for Geographic Information and Geovisualization*, 10 (2), 112–122. doi:10.3138/FM57-6770-U75U-7727

Duda, R.O. and Hart, P.E., 1972. Use of the Hough transformation to detect lines and curves in pictures. *Communications of the ACM*, 15 (1), 11–15. doi:10.1145/361237.361242

Eddy, W.F., 1977. A new convex hull algorithm for planar sets. *ACM Transactions on Mathematical Software*, 3 (4), 398–403. doi:10.1145/355759.355766

Edelsbrunner, H., Kirkpatrick, D., and Seidel, R., 1983. On the shape of a set of points in the plane. *IEEE Transactions on Information Theory*, 29 (4), 551–559. doi:10.1109/TIT.1983.1056714

Edelsbrunner, H. and Mücke, E.P., 1994. Three-dimensional alpha shapes. *ACM Transactions on Graphics*, 13 (1), 43–72. doi:10.1145/174462.156635

Ester, M., *et al*., 1996. A density-based algorithm for discovering clusters in large spatial databases with noise. *In*: E. Simoudis, J. Han, and U. Fayyad, eds. *Proceedings of the second international conference on Knowledge Discovery and Data mining (KDD-96)*, 2–4 August 1996 Portland. Menlo Park, CA: AAAI Press, 226–231.

Estivill-Castro, V., 2002. Why so many clustering algorithms: a position paper. *ACM SIGKDD Explorations Newsletter*, 4 (1), 65–75. doi:10.1145/568574.568575

Filin, S. and Pfeifer, N., 2006. Segmentation of airborne laser scanning data using a slope adaptive neighborhood. *ISPRS Journal of Photogrammetry and Remote Sensing*, 60 (2), 71–80. doi:10.1016/j.isprsjprs.2005.10.005

Fischler, M.A. and Bolles, R.C., 1981. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24 (6), 381–395. doi:10.1145/358669.358692

Graham, R.L., 1972. An efficient algorith for determining the convex hull of a finite planar set. *Information Processing Letters*, 1 (4), 132–133. doi:10.1016/0020-0190(72)90045-2

Grigillo, D. and Kanjir, U., 2012. Urban object extraction from digital surface model and digital aerial images. *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*, I-3, 215–220. doi:10.5194/isprsannals-I-3-215-2012

Haala, N. and Kada, M., 2010. An update on automatic 3D building reconstruction. *ISPRS Journal of Photogrammetry and Remote Sensing*, 65 (6), 570–580. doi:10.1016/j.isprsjprs.2010.09.006

Hellerstein, J.M., Naughton, J.F., and Pfeffer, A., 1995. Generalized search trees for database systems. *In*: U. Dayal, P.M.D. Gray, and S. Nishio, eds. *Proceedings of the 21th international conference on very large data bases*, 11–15 September 1995 Zurich, Switzerland. San Francisco, CA: Morgan Kaufmann Publishers Inc, 562–573.

Henn, A., *et al*., 2013. Model driven reconstruction of roofs from sparse LIDAR point clouds. *ISPRS Journal of Photogrammetry and Remote Sensing*, 76, 17–29. doi:10.1016/j.isprsjprs.2012.11.004

Huang, H., Brenner, C., and Sester, M., 2013. A generative statistical approach to automatic 3D building roof reconstruction from laser scanning data. *ISPRS Journal of Photogrammetry and Remote Sensing*, 79, 29–43. doi:10.1016/j.isprsjprs.2013.02.004

Jarvis, R.A., 1973. On the identification of the convex hull of a finite set of points in the plane. *Information Processing Letters*, 2 (1), 18–21. doi:10.1016/0020-0190(73)90020-3

Kim, C., *et al*., 2016. Segmentation of planar surfaces from laser scanning data using the magnitude of normal position vector for adaptive neighborhoods. *Sensors*, 16 (2), 140. doi:10.3390/s16020140

Kwak, E. and Habib, A., 2014. Automatic representation and reconstruction of DBM from LiDAR data using recursive minimum bounding rectangle. *ISPRS Journal of Photogrammetry and Remote Sensing*, 93 (7), 171–191. doi:10.1016/j.isprsjprs.2013.10.003

Lari, Z. and Habib, A., 2014. An adaptive approach for the segmentation and extraction of planar and linear/cylindrical features from laser scanning data. *ISPRS Journal of Photogrammetry and Remote Sensing*, 93 (7), 192–212. doi:10.1016/j.isprsjprs.2013.12.001

Leberl, F., *et al*., 2010. Point clouds: LiDAR versus 3D vision. *Photogrammetric Engineering Remote Sensing*, 76 (10), 1123–1134. doi:10.14358/PERS.76.10.1123

Mandal, D.P. and Murthy, C.A., 1997. Selection of alpha for alpha-hull in $\mathbb{R}^2$. *Pattern Recognition*, 30 (10), 1759–1767. doi:10.1016/S0031-3203(96)00176-8

Melkemi, M. and Djebali, M., 2001. Weighted A-shape: a descriptor of the shape of a point set. *Pattern Recognition*, 34 (6), 1159–1170. doi:10.1016/S0031-3203(00)00063-7

Moreira, A. and Santos, M.Y., 2007. Concave hull: a *k*-nearest neighbours approach for the computation of the region occupied by a set of points. *In*: J. Braz, P. Vázquez, and J.M. Pereira, eds. *Proceedings of the second international conference on computer graphics theory and applications*, 8-11 March 2007 Barcelona, Spain. Setubal, Portugal: Institute for Systems and Technologies of Information, Control and Communication, 61–68.

OGC (Open Geospatial Consortium), 2006. *OpenGIS® implementation standard for geographic information - simple feature access - part 1: common architecture*. OGC06-103r4. Available from: http://portal.opengeospatial.org/files/?artifact_id=25355

Preparata, F.P. and Hong, S.J., 1977. Convex hulls of finite sets of points in two and three dimensions. *Communications of the ACM*, 20 (2), 87–93. doi:10.1145/359423.359430

Rabbani, T., van Den Heuvel, F.A., and Vosselman, G., 2006. Segmentation of point clouds using smoothness constraint. *International archives of photogrammetry, remote sensing and spatial information sciences*, 36 (5), 248–253.

Ramer, U., 1972. An iterative procedure for the polygonal approximation of plane curves. *Computer Graphics and Image Processing*, 1 (3), 244–256. doi:10.1016/S0146-664X(72)80017-0

Rottensteiner, F., *et al*., 2014. Results of the ISPRS benchmark on urban object detection and 3D building reconstruction. *ISPRS Journal of Photogrammetry and Remote Sensing*, 93 (7), 256–271. doi:10.1016/j.isprsjprs.2013.10.004

Sampath, A. and Shan, J., 2007. Building boundary tracing and regularization from airborne LiDAR point clouds. *Photogrammetric Engineering & Remote Sensing*, 73 (7), 805–812. doi:10.14358/PERS.73.7.805

Sampath, A. and Shan, J., 2010. Segmentation and reconstruction of polyhedral building roofs from aerial LiDAR point clouds. *IEEE Transactions on Geoscience and Remote Sensing*, 48 (3), 1554–1567. doi:10.1109/TGRS.2009.2030180

Shewchuk, J.R., 2002. Delaunay refinement algorithms for triangular mesh generation. *Computational Geometry*, 22 (1–3), 21–74. doi:10.1016/S0925-7721(01)00047-5

Stroud, I., 2006. *Boundary representation modelling techniques*. London: Springer-Verlag London.

Toth, C. and Jóźków, G., 2016. Remote sensing platforms and sensors: a survey. *ISPRS Journal of Photogrammetry and Remote Sensing*, 115, 22–36. doi:10.1016/j.isprsjprs.2015.10.004

van Oosterom, P., *et al*., 2015. *Realistic benchmarks for point cloud data management systems*. Available from: http://www.gdmc.nl:8080/mpc/documents/papers/realistic-benchmarks-for-point-cloud-data-management-systems/view [Accessed 21 July 2016].

Verma, V., Kumar, R., and Hsu, S., 2006. 3D building detection and modeling from aerial LIDAR data. *In*: A. Fitzgibbon, C.J. Taylor, and Y. LeCun, eds. *2006 IEEE computer society conference on computer vision and pattern recognition*, 17-22 June 2006, New York, NY. IEEE, 2213–2220. doi:10.1109/CVPR.2006.12

Vosselman, G. and Dijkman, E., 2001. 3D building model reconstruction from point clouds and ground plans. *International Archives of Photogrammetry and Remote Sensing*, XXXIV-3/W4, 37–43.

Vosselman, G. and Gorte, B., 2004. Recognising structure in laser scanner point clouds. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, XXXVI-8/W2, 33–38.

Wang, J. and Shan, J., 2009. Segmentation of LiDAR point clouds for building extraction. *In*: *ASPRS 2009 annual conference*, 9-13 March 2009, Baltimore, Maryland. Available from: https://engineering.purdue.edu/~jshan/publications/2009/ASPRS_2009_Lidar.pdf

Xiong, B., Oude Elberink, S., and Vosselman, G., 2014. A graph edit dictionary for correcting errors in roof topology graphs reconstructed from point clouds. *ISPRS Journal of Photogrammetry and Remote Sensing*, 93, 227–242. doi:10.1016/j.isprsjprs.2014.01.007

Xiong, B., *et al*., 2015. Flexible building primitives for 3D building modeling. *ISPRS Journal of Photogrammetry and Remote Sensing*, 101, 275–290. doi:10.1016/j.isprsjprs.2015.01.002

Xu, L., Kong, D., and Li, X., 2014. On-the-fly extraction of polyhedral buildings from airborne LiDAR data. *IEEE Geoscience and Remote Sensing Letters*, 11 (11), 1946–1950. doi:10.1109/LGRS.2014.2314458

Zhang, K., Yan, J., and Chen, S.C., 2006. Automatic construction of building footprints from airborne LiDAR data. *IEEE Transactions on Geoscience and Remote Sensing*, 44 (9), 2523–2533. doi:10.1109/TGRS.2006.874137

Zhou, Q. and Neumann, U., 2008. Fast and extensible building modeling from airborne LiDAR data. *In*: *Proceedings of the 16th ACM SIGSPATIAL international conference on advances in geographic information systems*, 5-7 November 2008 Irvine, CA. New York, NY: ACM, 1–7. doi: 10.1145/1463434.1463444

Zhao, Z., *et al*., 2016. Extracting buildings from and regularizing boundaries in airborne LiDAR data using connected operators. *International Journal of Remote Sensing*, 37 (4), 889–912. doi:10.1080/01431161.2015.1137647