

Modelling of buildings from aerial LiDAR point clouds using TINs and label maps

Minglei Li^{a,b,*}, Franz Rottensteiner^b, Christian Heipke^b

^a College of Electronic and Information Engineering, Nanjing University of Aeronautics and Astronautics, Jiangjun St. 29, 211106 Nanjing, China

^b Institute of Photogrammetry and GeoInformation, Leibniz Universität Hannover, Nienburger Straße 1, 30167 Hannover, Germany



ARTICLE INFO

Keywords:

LiDAR
Urban modelling
TIN model
Graph cut
Normal vector guidance

ABSTRACT

This paper presents a new framework for automatically creating compact building models from aerial LiDAR point clouds, where each point is known to belong to the class *building*. The approach addresses the issues of non-uniform point density and outlier detection to extract and refine semantic roof structures by a sequence of operations on a label map. We first partition the points into some coarse regions based on a region growing method over the Triangulated Irregular Network (TIN) model. The region label IDs are then projected to a 2D grid map, which is used to refine the roof regions and their boundaries. We design an energy optimization approach on the label map to optimize the region labels. In order to regularize the contours of roof regions extracted from the label map, we propose a new method for refining contour segment vertices, which iteratively filters the normals of contour segments and uses them to guide the update of contour vertices. The effectiveness of this method is evaluated on LiDAR point clouds from different scenes, and its performance is validated by extensive comparisons to state-of-the-art techniques.

1. Introduction

Airborne light detection and ranging (LiDAR) provides an efficient and reliable way to survey large scale urban scenes. Extracting buildings from LiDAR point clouds is crucial for many applications, such as urban planning, emergency response, and vehicle navigation. A basic characteristic of aerial LiDAR data is that information on roof structures of buildings is present in the data, but wall information is incomplete or missing.

Recent works (Lafarge and Mallet, 2012; Serna and Marcotegui, 2014; Weinmann et al., 2015a, 2015b; Blomley et al., 2016; Niemeyer et al., 2016) have introduced successful approaches to extract points belonging to the class *building* from massive LiDAR point clouds. Based on this foundation, many different approaches have been proposed for creating polyhedral building models (Musalski et al., 2013; Wang et al., 2018, see also the overview in Rottensteiner et al., 2014). Nevertheless, some problems with building model creation still exist. The first challenge stems from the fact that the raw data obtained from airborne LiDAR usually exhibit noise and non-uniform point densities because of different observation distances, object reflectance, and overlaps. Secondly, the aerial LiDAR data are unstructured, which leaves an unclear definition of neighbourhood relationships between points. Moreover, since the buildings in different cities often have

different shapes, widely applicable patterns do not exist. These characteristics of point clouds make it difficult to extract stable features or widely applicable patterns from the raw data. As a result, most existing methods are only valid for certain data, but have difficulties to adapt to a large variety of different data.

This paper presents a new framework for automatically creating compact building models from aerial LiDAR point clouds, where each point is known to belong to the class *building*. The approach addresses the issues of non-uniform point density and outlier detection to extract and refine semantic roof structures by a sequence of operations on a 2D label map. Automation is one of the aims of our work; another key point is adaptation to complex boundaries of different buildings. It is assumed that the airborne data reflect the 2.5 dimensional characteristics of the scene, so we use the 2D label map as intermediate representation to accelerate the procedure and to refine the model.

The main contributions of this work are twofold:

- In order to recover the roof structures, we link a Triangulated Irregular Network (TIN) model of point clouds to a label map of roof regions. The initial roof structures are extracted from the TIN model and transformed to a grid-based 2D label map. Then, the graph cut method is applied to that label map to optimize the roof structures.
- After defining normal vectors for roof contour segments in the label

* Corresponding author.

E-mail address: minglei_li@126.com (M. Li).

map, we present a filtering method to adjust the normals, and these normals are further used to guide an iterative refinement of the contour vertices.

The proposed algorithm is a fully automated method that generates watertight polyhedral models of buildings or building blocks from aerial LiDAR point clouds.

2. Related work

Over the past few decades, a wealth of research has appeared to address the problems of urban modelling. The data are obtained by satellite, aircraft or ground equipment, and the data types vary from images to laser points. In this section, we mainly discuss the works of modelling urban buildings from aerial LiDAR point clouds. The whole process can be executed in 3D or on a 2D support plane. 2D support grids for 3D point processing have been introduced in the past (Miliareis and Kokkas, 2007; Meng et al., 2009; Lafarge and Mallet, 2012; Sun and Salvaggio, 2013; Chen et al., 2014; Li et al., 2015). These methods project 3D points to grid cells and carry out further processing; other approaches work directly on the 3D point cloud (Overby et al., 2004; Dorninger and Nothegger, 2007; Verma et al., 2006; Oude Elberink and Vosselman, 2009; Xiong et al., 2015; Nan and Wonka, 2017). In general, no matter whether an approach is data- or model-driven, most methods can be decomposed into three components: roof primitive detection, contour refinement, and model generation. These three components are closely related.

Roof primitive detection. In order to obtain roof primitives, region growing based approaches are often used; they are easy to implement and efficient for data of more or less equal point density and noise level. When applying region growing, most efforts are directed towards seed selection and criteria for region extension. The similarities of point normal vectors or surface curvatures are the most common criteria for grouping points (Sun and Salvaggio, 2013; Chen et al., 2017). Dorninger and Nothegger (2007) determined seed-clusters in feature spaces of plane parameters by a clustering method, then performed a region growing for each seed-cluster. Chen and Zhao (2012) used a hybrid way to select seed points by combining LiDAR points and the corresponding images. Nurunnabi et al. (2012) refined the seed points by removing points along sharp edges using a curvature threshold. Vo et al. (2015) proposed using an octree segmentation to generate a coarse voxel-based partition, which determines the termination of partitions based on a residual threshold and a minimum voxel size. Then, they merge coplanar voxels by voxel-based region growing and refine building edge areas by point-based region growing. The selection of seed points strongly affects the segmentation quality, and a universally valid criterion does not exist. When point density is non-uniform, an adaptive neighbourhood needs to be defined for region growing-based methods, because a fixed neighbourhood can lead to disconnected regions, if selected too small.

Another common way to detect roof primitives is known as parametric model fitting. These methods use parameterized models to fit the observed scenes. Two widely employed approaches are the Hough Transform (Overby et al., 2004; Vosselman et al., 2004) and random sample consensus (RANSAC) (Fischler and Bolles, 1981; Tarsha-Kurdji et al., 2007; Schnabel et al., 2007; Henn et al., 2013). These methods iteratively detect potential geometric items which can be parameterized, such as cylinders, planes, and spheres. The outcomes have compact forms and are relatively invariant to noise and outliers. However, when the scene is large and the point density is non-uniform, finding thresholds that are adaptive to the data, is difficult.

Contour refinement. After detecting roof primitives, the raw contours of those primitives usually consist of noisy polygons, so contour refinement is required. Most methods apply collinearity or rectangularity constraints. In order to extract the requested results, Meng et al. (2009) and Yang et al. (2013) proposed to resample LiDAR points into

grids and then to extract building outlines by morphological operators. Chen et al., (2017) clustered the points using an enhanced probability density clustering algorithm, then constructed Delaunay triangles from labelled roof points and finally calculated the roof primitive boundaries based on a sub-graph of the Voronoi diagram. To simplify the contour shapes, the Douglas-Peucker algorithm is frequently used as a pre-process to reduce the number of vertices (Zhang et al., 2006; Poullis and You, 2009; Sohn et al., 2012; Li et al., 2015; Xiong et al., 2016). After obtaining simplified segments, different methods use various strategies to regularize the contour shapes. Zhou and Neumann (2008) determined the buildings' dominant directions through statistical analysis of the boundary segments, and then regularized the boundary segments along these directions. Similarly, Albers et al. (2016) used the Hough Transform to extract line segments and determine the main directions of the building, and then regularized the boundary segments using direction-constrained energy minimization. However, some buildings do not only have one main direction.

If the neighbouring roof planes are intersected, the mutual boundaries can be directly determined by intersecting the planar patches (Rottensteiner, 2006; Sampath and Shan, 2010; Lafarge et al., 2010); note, that this method is not applicable to step boundaries. Rottensteiner et al. (2005) proposed a method to detect step edges based on domain specific information, and adjusted the vertices of step contours by a least squares method. Perera et al. (2014) proposed to determine the positions of boundaries by integrating the edges extracted from aerial images. However, such corresponding aerial images are not always available. To eliminate erroneous vertices and regularize contour shape, Sohn et al. (2012) proposed a global optimization method to minimize a description length model which combines terms of model approximation and boundary complexity. Xiong et al. (2016) proposed to partition building data into different layers based on the roof heights, then the contours of each layer are snapped to the contours of a known footprint map of the building. This method requires a known 2D map containing footprints of buildings; generating such a footprint map may be a non-trivial work.

Model generation. In order to generate compact polyhedral models, Poullis and You (2009) used local neighbourhood information to recover missing data and fill in potential holes, and they reconstructed dome-like and stadium-like structures by fitting ellipsoidal primitives. Zhou and Neuman (2010) developed a dual contouring modelling method to simplify the triangle meshes. In a follow-up (Zhou and Neumann, 2012) through extracting minimal bounding boxes of roof patches, topology control and global regularity were considered to improve the quality of the boundaries. Lafarge and Mallet (2012) used previously detected geometric primitives to represent irregular roof components. The selection of these primitives was cast as a non-convex energy minimization problem. These methods are basically data-driven; how to determine the proper parameters for data-driven algorithms is a difficult problem.

Another category for model generation is known as pattern-driven methods. Some urban buildings exhibit repetitive structures or regular shapes, so pattern-based modelling methods can approximate these data by fitting predefined building structure patterns. Lafarge et al. (2010) and Lin et al. (2013) built libraries of predefined blocks and transformed these blocks onto the 2D-support or the decomposed points via energy optimization which control the fitting errors. Li et al. (2016a) used the assumption of a Manhattan-world to obtain an approximate reconstruction of the buildings by fitting boxes directly into the point clouds. The roof topology graph has been proposed as a special representation of the inner structure of building roofs without considering their geometric positions (Verma et al., 2006; Oude Elberink and Vosselman, 2009; Xiong et al., 2015). A so-called graph edit dictionary records some common topological relationships. Elements of this dictionary are assembled by graph matching to best represent the data. In order to further correct topological errors in the roof topology graph, Xiong et al. (2014) proposed a dictionary of graph

edit operations to automatically identify and correct the errors. [Wichmann et al. \(2018\)](#) presented a so-called RoofN3D network using machine learning to distinguish between different roof types and to recognize certain roof structures. In general, pattern-driven methods usually need a library of known roof structure patterns, but the building shapes in reality are typically more variable and can therefore not be fully covered. Besides, pre-processing, such as primitive detection and intersection line detection, is usually necessary for pattern-driven methods.

Mathematical optimization is frequently used in many modelling methods. For instance, [Sohn et al. \(2012\)](#) used a gradient descent method to minimize their description length model. [Verdie et al. \(2015\)](#) first initialized a dense mesh representation, and then extracted a surface model from the mesh using a min-cut algorithm. [Poullis \(2013\)](#) simplified the boundaries of roof patches using a graph cut based energy minimization process. [Li et al. \(2016b\)](#) decomposed the scene by a set of boxes under the assumption of a Manhattan-world. Then, they represented the box candidates as nodes in a graph and used graph cut to select a set of valid boxes to approximate the geometry of the buildings. [Nan and Wonka \(2017\)](#) generated a number of planar patches by intersecting all detected plane items and then selected a subset of patches to generate the model employing linear programming. For 3D analysis, the main challenge of mathematical optimization is how to calculate the data terms and define the energy formulations. Besides, computational complexity is an important issue for large scale optimization.

Our modelling approach is closely related to the primitive arrangement approach of [Lafarge and Mallet \(2012\)](#). The similarity is that both methods use a graph cut approach to optimize the delineation of roof elements initialized on a 2D supporting grid. However, our method differs from [Lafarge and Mallet \(2012\)](#) in the following ways. Firstly, we divide the facets of a Triangulated Irregular Network (TIN) model into different roof regions, and then project the initial roof label IDs of the triangle facets onto the supporting grid rather than the discrete 3D points. Point-based projection may result in some empty cells or redundant points in cells depending on the point density. On the contrary, as a TIN model is a continuous surface, it is ensured that every grid cell obtains exactly one initial roof label ID from the triangle facets. In this way, our method is more adaptive to a range of grid widths and non-uniform point density. Secondly, we introduce a novel normal vector-guided method for roof contour refinement. We first filter the segment normal vectors, and then use these normals to guide the adjustment of the contour vertices based on a soft angle regularization criterion. Our approach represents a general mathematical formulation for contour segment arrangement problem.

While in general gridding data comes with the danger of losing information, there are two main advantages of using a grid-based data structure to process LiDAR points. Firstly, rasterization converts irregularly spaced points into regular structure with standard neighbourhood relationships, e.g. 4- or 8-connected neighbourhood. We can establish and track region contours more easily on the 2D grid with such neighbourhood relationship. Secondly, as gridded data have a fixed grid size and thus uniform grid point distribution, grid-based methods can reduce the dependence of models on point cloud density by projecting region label IDs to cells, and control the precision of models by setting the grid width. The danger of information loss is counteracted by choosing a suitably small grid size.

3. Proposed method

3.1. Overview

The workflow of our new approach is designed as a three-stage process, as shown in [Fig. 1](#).

(i) Firstly, given an aerial LiDAR point cloud, where each points

belongs to the class *building*, we build TIN models from the data. A region growing method is developed to extract locally coherent planar regions by traversing through the triangle facets. Then, a trimming process is used to delete invalid regions. As a result, each region is assumed to be related to a planar roof primitive encoded with a region label ID.

- (ii) We generate a 2D supporting label map represented as a grid, whose initial cell values are obtained by projecting the region label IDs from the TIN facets to the grid. A graph cut approach is applied to the label map to refine the label assignment. After optimization, the regions in the map are assumed to provide a good representation of the roof faces, and the region contours to correspond to the projection of the patch contours.
- (iii) The contours are extracted from the label map and refined. We simplify the contour chains by the Douglas-Peucker algorithm followed with a segment normal-guided refinement for contour vertices. The refinement drives the roof contours to satisfy soft collinearity and rectangularity constraints. Finally, the building models are generated by connecting roof patches and the ground along vertical planes.

3.2. Triangulation based region growing

Based on the assumption that roof patches are piecewise planar, we decompose the aerial LiDAR point cloud into independent regions which belong to different roof planes. In order to handle the problem of non-uniform point density and noise, we convert the points to a TIN model by Delaunay triangulation and use a TIN-based region growing algorithm to extract planar primitives, which guarantees that every cell in the label map has exactly one region label ID corresponding to the planar primitives.

For region growing we traverse all triangles using common edges; neighbouring triangles are merged, if their normal vectors point in a similar direction. To deal with noise in the TIN models, a median filter is applied to first smooth the triangle facet normal vectors. Once a triangle is added to a group, the median normal vector of that group is recomputed. A group stops extending when no more triangles satisfy the merging conditions, and a new grouping starts at the next ungrouped neighboured triangle. The procedure continues in the same fashion until all triangles are visited. After region growing, every triangle has a region label ID. Compared with point-based region growing methods, the TIN-based method does not need an explicitly defined neighbourhood radius which is difficult to set with non-uniform point density.

The region growing results may contain some regions which do not correspond to a valid roof primitive, because the raw point cloud may exhibit outliers and gaps, which seriously degenerate the TIN model, as shown in [Fig. 2\(a\)](#). We use a trimming strategy to deal with this problem, in which two types of regions are eliminated. As shown in [Fig. 2\(b\)](#), (i) the first type are those with nearly horizontal normal vectors, i.e. almost vertical triangles; (ii) the second type are those whose area projected to the ground plane is smaller than a threshold A , meaning they are too small to represent a valid roof. In [Fig. 2\(b\)](#), the deleted triangles are marked in grey, and the valid roof regions are highlighted by random colours. For each region a plane Π is then calculated by least squares fitting from all related triangle vertices. Each triangle is finally associated with an integer region label ID, which is used as initial label ID in the following optimization.

3.3. Label map generation

For generating the label map, similarly to other researchers, we use 2D support grids (see [Section 2](#)). In contrast these approaches, however, our method projects the initial roof label IDs of the triangle facets to the grid cells. As the TIN model covers the entire area, the projection process thus ensures that every cell obtains exactly one label ID.

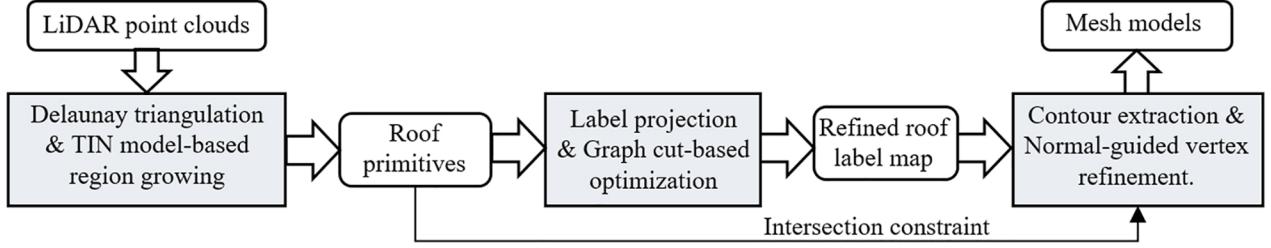


Fig. 1. The workflow of the proposed approach.

As mentioned in Section 3.2, we deleted a number of undesired triangles, so the label map contains some cells which are not assigned to any valid region, such as the unascertained cells shown in Fig. 3. In order to obtain a watertight model, we need to remove these gaps; we do so using energy optimization. Firstly, the label map is associated with a regular graph $G = (V, E)$, where $V = \{v_1, v_2, \dots, v_m\}$ denotes the set of nodes relating to the cells, and $E = \{e_1, e_2, \dots, e_n\}$ denotes the set of edges relating to the connections between all pairs of adjacent cells under a standard 8-connected raster neighbourhood topology. The refinement is a multi-labelling procedure that assigns every cell to either one of the roof region label IDs or the ground label ID. The joint labelling assignment $L(V)$ is calculated by minimizing the objective function $En(L)$, which includes a data cost and a smoothness cost:

$$En(L) = \sum_{v \in V} D_v(l_v) + \sum_{e_{p,q} \in E} S_{e_{p,q}}(l_p, l_q) \quad (1)$$

The data cost $D_v(l_v)$ checks the coherence of the label l_v at the cell node v with respect to the input data. $S_{e_{p,q}}(l_p, l_q)$ is a smoothness term; $e_{p,q}$ denotes an edge connecting nodes p and q . If the number of roof primitives is M , including the ground label ID, there are $M + 1$ possible label IDs.

To analyse the coverage of roof regions and to handle holes within roof regions, we use the 2D alpha shape of cells having the same initial label ID. For a cell group $\{v^l_i\}$ marked with l_i , we compute its alpha shape α^l_i which forms a closed region containing the cells in the group. α^l_i is represented by a set of contour points $\{p_{\text{contour}}\}$. The ground plane is treated as a special group, whose alpha shape is defined to coincide with the alpha shape of the entire TIN model, because the ground region is the complement of the region of the entire TIN model in the label map.

Then, the data term is computed based on the relationship between the cells and the alpha shapes. If a cell c lies inside the alpha shape α^l_i of the region group with label l_i , the data cost for assigning c to label l_i is 0. Otherwise, the data cost is calculated based on two terms including both horizontal and vertical distance metrics: (i) the horizontal distance $d(c, \alpha^l_i)$ from c to α^l_i , which is approximated by the minimal distance from c to the contour points of the alpha shape, i.e. $\{p_{\text{contour}}\}$; (ii) the height difference $d(c, \Pi^l_i)$ between the TIN model and plane Π^l_i at position (x_c, y_c) . The data cost for assigning c to a label l_i then reads:

$$D_c(l_i) = \begin{cases} 0 & \text{if } c \text{ lies inside } \alpha^l_i \\ d(c, \alpha^l_i) + d(c, \Pi^l_i) & \text{otherwise} \end{cases} \quad (2)$$

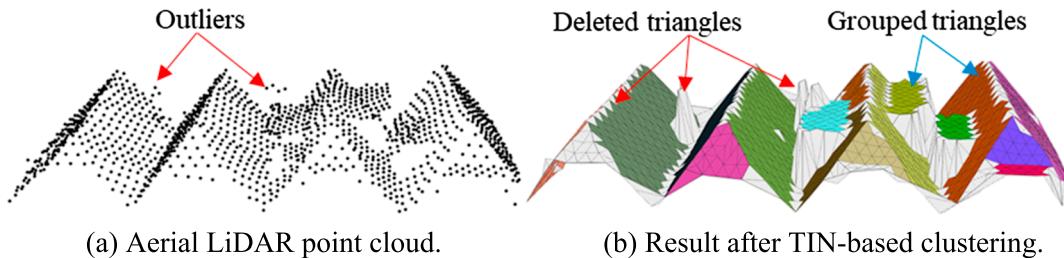


Fig. 2. TIN-based region growing.

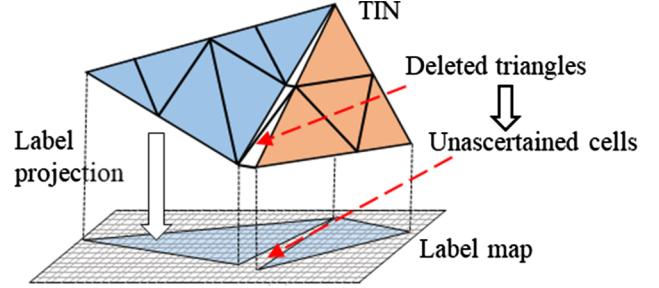


Fig. 3. Projection of the region labels from the TIN model to the label map.

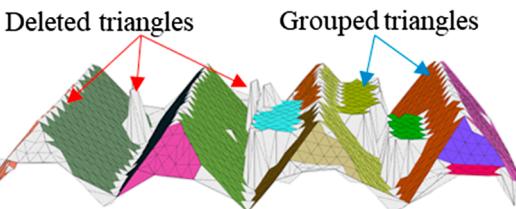
$$\begin{aligned} d(c, \alpha^l_i) &= \min_{p \in \alpha^l_i} \|c - p\| \approx \min_{p' \in \{p_{\text{contour}}\}} \|c - p'\| \\ d(c, \Pi^l_i) &= |h(c) - h(c^{\Pi^l_i})| \end{aligned}$$

where $h(c)$ and $h(c^{\Pi^l_i})$ denote the height of the TIN model and the height of plane Π^l_i at (x_c, y_c) , respectively.

We have also tried a simpler approach to calculate the data term which only uses the perpendicular distance from the candidate label plane to the raw data, such as d_i for label i and d_j for label j as shown in Fig. 4. However, this approach turned out to be unreliable: as the gaps in the label map are mainly caused by outliers in the raw data, the perpendicular distance may not reflect the trend of the roof structure correctly, especially when some roof planes are steep. For example in Fig. 4, $h(c)$ at c is far from both planes Π^i and Π^j . If we use the plane projection distances, the result tends to assign c to label j because $d_i > d_j$, while cell c is closer to the region of label i . On the contrary, the cost of our data term in (2) for assigning label i to c is $d(c, \alpha^i) + d(c, \Pi^i)$, which is smaller than that for assigning label j , $d(c, \alpha^j) + d(c, \Pi^j)$.

In (1), the smoothness term $S_{e_{p,q}}(l_p, l_q)$ is designed to encourage spatial coherence by penalizing discontinuities between neighbouring cells. It is assumed that label discontinuities exist at the boundaries of adjacent regions. In order to enhance the ability to classify the cells near the boundaries, we introduce a binary feature vector f that encodes the positional characteristic of every cell. Each binary element in f is determined by the positional relationship between the cell and the region intersection lines.

Firstly, the intersection lines of all pairs of roof plane primitives are calculated and projected to the label map. Only when the minimum



(b) Result after TIN-based clustering.

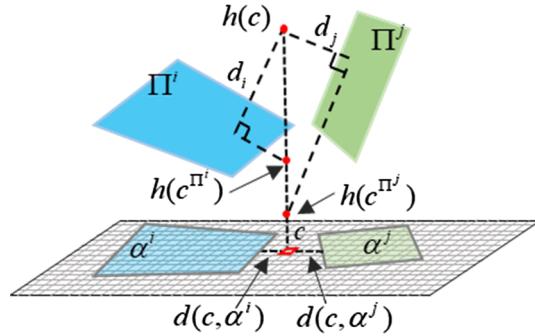


Fig. 4. An illustration of the data term composition.

distance between the alpha shapes of two planes is smaller than a threshold, their intersection line is taken into account. In the label map, each line divides the 2D plane into two half-spaces, **positive** and **negative**, which correspond to codes 1 and 0. An example is shown in Fig. 5. If there are m intersection lines, the dimension of the binary vector f is m . When a cell lies on the positive side of the k^{th} line, the k^{th} element of f is 1, and otherwise its value is 0. Subsequently, given two neighbouring cells c_p and c_q , the smoothness term between them is defined as:

$$S_{p,q}(l_p, l_q) = \begin{cases} 0 & \text{if } l_p = l_q \\ 2w & \text{if } l_p \neq l_q \text{ AND } f_p = f_q \\ w & \text{otherwise} \end{cases} \quad (3)$$

Here, if c_p and c_q are assigned the same label, i.e., $l_p = l_q$, the smoothness cost is 0. If $f_p = f_q$, the two cells are more likely to be located in the same region, so a larger cost must be paid when assigning them different labels. For other cases, we use a base smoothness cost w , which is set to the average width of the roofs. As we can obtain the area of every roof region from the TIN model, w is approximated as the square root of the average area of all roofs.

The energy function (1) is minimized using graph cut (Kolmogorov and Zabih, 2004). As a result, the label map represents the roof structures and the ground in a seamless form. An illustration of the optimization effect is presented in Fig. 6. As shown in the enlarged parts in Fig. 6(b) and (c), using feature vector f in the smoothness term significantly improves the labelling result at the boundaries.

3.4. Contour extraction and model meshing

After optimizing the roof regions, the contours of these roof regions in the label map may still have irregular shapes. As most common buildings have polygonal outlines, it is reasonable to simplify the model boundaries by a set of linear segments. The following steps describe the method to produce contour vertices for each roof.

To extract the roof contours, we turn to the corner points of cells in the label map. These corner points are classified into three categories based on the number of label ID types in the surrounding cells. An illustration is shown in Fig. 7. The first kind of corner points are the

junction points, which are surrounded by cells assigned to three or four labels. The second kind of corner points are the **contour points** with exactly 2 types of cells. The third kind of corner points are **non-contour points** which are surrounded by cells all having the same label ID. In this way, the junction points partition the contour of a roof into a sequence of **chains**, each of which consists of two junction points and some contour points between them. A closed polygon roof is a special case where the contour consists of only contour points without any junction points.

Every chain is firstly simplified by the Douglas-Peucker (D-P) algorithm (Douglas and Peucker, 1973). For example, in Fig. 7(b), the number of contour points after simplification is decreased from 17 to 1, while the number of junction points has not changed.

Next, we categorize the simplified chains into two types, **intersection chain** and **step chain**, based on their relationship to the adjacent roofs. If the adjacent roof planes intersect at their connected region border, their shared chain is an intersection chain. Otherwise, the shared chain is a step chain. The chain type is determined as follows. Given a chain $\phi: \{p_1, \dots, p_n\}$, which consists of some vertices p_i , if the sum of the distances between all vertices to the intersection line of the chain's neighbouring roof planes is smaller than a threshold, this chain is considered an intersection chain, otherwise a step chain. When dealing with an intersection chain, we directly project the contour points to the intersection lines to obtain updated vertices.

Step chains are common at building boundaries because they are located between roofs and the ground. Some buildings with horizontal rooftops or dormers also have internal step chains. Since collinear and rectangular polygonal contours are very common in the real world, we investigate whether the step chains have such specific shapes. If so, we regularize the chain based on soft collinearity and rectangularity constraints.

Given a step chain ϕ with vertices $\{p_1, \dots, p_i, p_j, \dots, p_n\}$, let ij denote a segment with end points p_i and p_j . The normal vector of a segment is defined as a unit vector perpendicular to the segment. For example, in Fig. 8(a), n_{ab} is the normal vector of segment ab whose endpoints are a and b . We use the refined segment normals to guide the vertex update iteratively.

According to the angle relationship between adjacent segments, the segment normals are filtered first. The filter is defined as a length weighted smoothing function, whose input are normal vectors of all segments connected to the target segment. As shown in Fig. 8(a), normal n_{bc} of segment bc is smoothed under the influence of n_{ab} , n_{ce} , and n_{cd} . For adjacent segments nearly collinear to bc , such as ab and cd , the normals are used directly in the smoothing filter. For adjacent segments nearly perpendicular to bc , such as ce , the normals are first rotated by 90° . In this way, the filter for normal n_{ij} reads as follows:

$$n'_{ij} = \frac{\ell_{ij} n_{ij} + \sum_{mn \in N_{ij}} \ell_{mn} F(n_{mn}, n_{ij})}{\| \ell_{ij} n_{ij} + \sum_{mn \in N_{ij}} \ell_{mn} F(n_{mn}, n_{ij}) \|} \quad (4)$$

$$F(n_{mn}, n_{ij}) = \begin{cases} \text{sgn}(n_{ij} \cdot n_{mn}) \cdot n_{mn} & \text{if } \cos\theta < \text{abs}(n_{ij} \cdot n_{mn}) \leqslant 1 \\ \text{sgn}(n_{ij} \cdot R_{90} n_{mn}) \cdot R_{90} n_{mn} & \text{if } 0 \leqslant \text{abs}(n_{ij} \cdot n_{mn}) < \sin\theta \\ 0 & \text{otherwise} \end{cases}$$

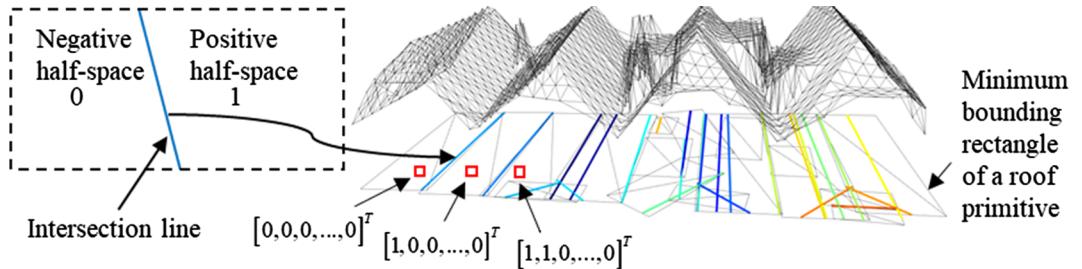


Fig. 5. The binary features of cells.

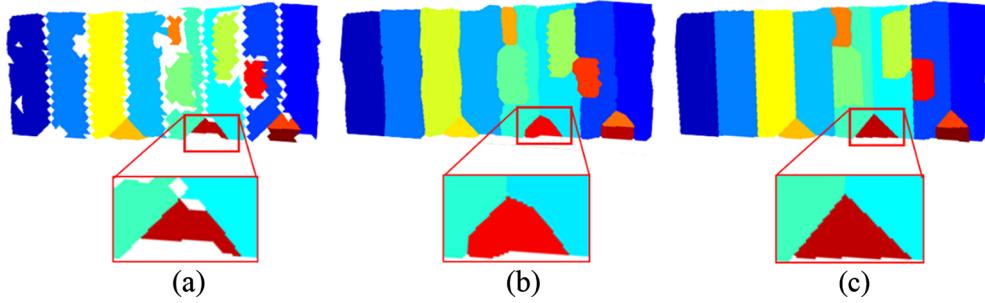


Fig. 6. The label maps of roof regions. (a) Initial label map; (b) optimization result without considering the condition of binary features (i.e. $2w$ in (3)); and (c) optimization result of the proposed method.

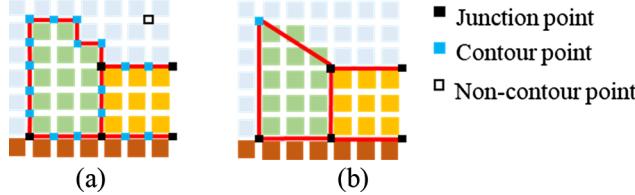


Fig. 7. An illustration of Douglas-Peucker based contour simplification.

where $mn \in N_{ij}$ denotes a segment mn adjacent to segment ij ; ℓ_{ij} and ℓ_{mn} are the segment lengths. $F(n_{mn}, n_{ij})$ is used to adjust the normal orientation by analysing the dot product of n_{mn} and n_{ij} . $F(n_{mn}, n_{ij})$ ensures that the influence of n_{mn} on n_{ij} is always non-negative, sgn stands for the sign function, and R_{90} rotates the normal n_{mn} by 90° . The condition $\cos\theta < \text{abs}(n_{ij} \cdot n_{mn}) \leq 1$ corresponds to the case of collinear, and $0 \leq \text{abs}(n_{ij} \cdot n_{mn}) < \sin\theta$ corresponds to that of perpendicular influence. θ is an angle threshold to determine whether the adjacent segments are nearly collinear, nearly perpendicular, or irrelevant. An illustration of possible relationships between n_{ij} and n_{mn} is given in Fig. 9.

Obviously, the dot product of a segment vector and its normal vector is 0. However, after normal vector filtering, the orientations of normal vectors are changed, so the sum of those dot products is no longer zero. Hence, we update vertices to minimize that sum:

$$\min_p \sum_{N_p} \sum_{pj} n'_{pj} \cdot (m_{pj} - p) \quad (5)$$

where m_{pj} is the centre of segment pj . In order to adjust the segments to match their new normals, the position of each vertex p is updated via:

$$p' = p + \frac{1}{\#N_p} \sum_{pj \in N_p} n'_{pj} [n'_{pj} \cdot (m_{pj} - p)] \quad (6)$$

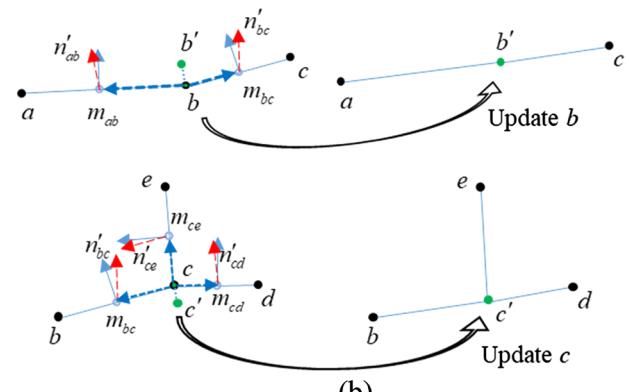
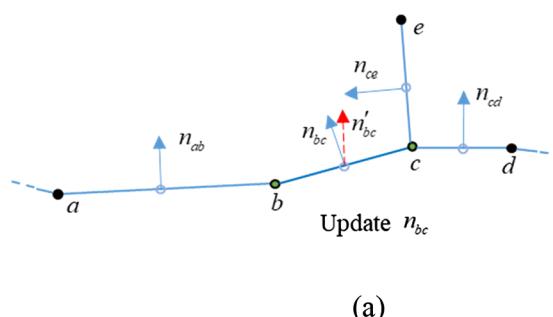


Fig. 8. The mechanism of segment normal-guided refinement for the contour vertices. (a) Filter example of n_{bc} affected by n_{ab} , n_{ce} , and n_{cd} . The red arrows represent the refined normal vector. (b) Normal-guided update for vertices b and c affected by refined normals. m_{ab} , m_{bc} , m_{cd} , and m_{ce} are the centres of the corresponding segments. Points b' and c' represent updated positions. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

where $\#N_p$ denotes the number of segments in N_p ; $\{pj\}$, which have p as one of the endpoints; n_{pj} is the normal vector of segment pj ; and m_{pj} is the centre point of segment pj . This scheme is actually a gradient descent process for (5). Note that the orientations of segment normals n_{pj} in (6) do not affect the result, because $n_{pj}'[n_{pj}' \cdot (m_{pj} - p)] = (-n_{pj}')[-(-n_{pj}') \cdot (m_{pj} - p)]$. Fig. 8(b) depicts an example of normal-guided vertex update for b and c .

According to (4) and (6), we iterate the three steps of segment normal initialization, filtering, and updating vertices until convergence is reached. The convergence criterion is that for every vertex the update is smaller than the grid size of the label map.

Finally, all contour vertices are back-projected from the label map to roof planes to calculate 3D positions. For completing the polyhedral model, the building walls are formed by vertically extruding the boundary contour vertices to adjacent roof layers or the ground. The refined roof contours and the modelling result corresponding to Fig. 2 and Fig. 6 are presented in Fig. 10. As shown in Fig. 10(a)–(c), the roof contours are evolved from coarse to fine by the Douglas-Peucker algorithm and the normal-guided vertex refinement. In (b) and (c), the dots depict the positions of segment endpoints and each contour chain is shown with a unique colour. In the mesh model, see Fig. 10(d), the walls are represented in grey and each rooftop is highlighted with a random colour.

4. Experimental results

4.1. Test data and experiment setup

We have tested our approach on different datasets for both qualitative and quantitative evaluation. The first test point clouds (**AHN3 dataset**) are provided by the Dutch National Cadastre ([AHN3, 2018](#)).

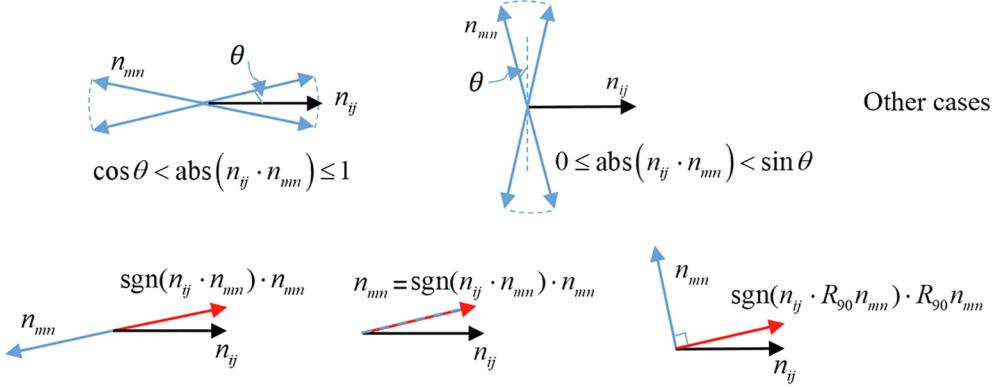


Fig. 9. $F(n_{mm}, n_{ij})$ ensures that the influence of n_{mm} on n_{ij} is non-negative.

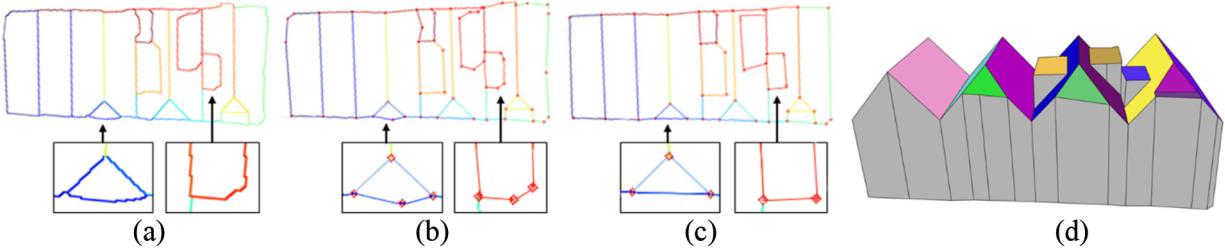


Fig. 10. The contour refinement results and a final mesh model. (a) Initial contours from the label map; (b) result of simplification; (c) result of segment normal-guided vertex refinement; and (d) polyhedral model.

This dataset is obtained by laser altimetry; the selected test region covers an area of approximately 9.300 m^2 , in which a total of 74,000 points were acquired with an average density of 8 points/ m^2 . Each point in the dataset is associated with a semantic label indicating its physical category, so we can directly obtain the point clouds of single buildings. The next test dataset (Vaihingen dataset) includes three areas; it was captured over the city of Vaihingen, Germany, and was provided by the German Society for Photogrammetry, Remote Sensing and Geoinformation (DGPF) (Cramer, 2010). Our test areas are included in the ISPRS benchmark for “Test Project on Urban Classification and 3D Building Reconstruction”. The data was captured with a Leica ALS50 system with a mean flying height above ground of 500 m. The average point density is about 4 points/ m^2 (Haala et al., 2010; Rottensteiner et al., 2014). For this dataset we manually selected the point clouds belonging to buildings from the Vaihingen dataset. Besides, we used two additional point clouds of single buildings released together with the references of Lafarge and Mallet (2012) (INRIA dataset, 2012) and Zhou and Neuman (2010) for comparison with their modelling algorithms. On average the point density of these two datasets is about 4 points/ m^2 and 20 points/ m^2 , respectively. The above mentioned datasets were selected because they contain a variety of rooftop structures in different orientations, and are thus appropriate to assess the proposed algorithm.

Our implementation involves a set of parameters: the area threshold A for deleting small planar primitives, the grid width w of the label map, the Douglas-Peucker threshold ϵ , a distance threshold Δ for detection intersection contour chains, and the angle threshold θ of collinearity judgment. Based on the observation of the physical world, we set $A \in [0.5, 2] \text{ m}^2$; for a low resolution grid map we select a larger value for A and vice versa. Different values for the grid width w were investigated, we discuss the corresponding effect in Section 4.4. In the other experiments, we set $w = 2/\text{density}$ in m. In most cases, using $\epsilon \in [w, 3w]$ simplifies contours efficiently. Empirically, we set Δ to 0.8 m and set θ to 20° . The angle threshold for similarity of normals used in region growing is also set to 20° . In general, we observed that the results are rather tolerant with respect to selecting specific values.

4.2. Urban modelling results

The selected point clouds of the AHN3 dataset are shown in Fig. 11(a), where the red¹ points correspond to buildings. Fig. 11(b) and (c) demonstrate that for most buildings our method precisely extracts the roof primitives and generates compact 3D models for the scene. However, some small roof structures pose difficulties: if there are enough raw points to identify the existence of a valid small roof, but its complete contours are not adequately reflected in the data, our contour regularization strategy may lead to incorrect contours, because it does not use higher level a priori assumptions.

In Fig. 12, we show the modelling results of three areas from Vaihingen. Based on the reference files of the ISPRS benchmark, we use the metrics proposed by Rutzinger et al., (2009) to evaluate the results. The analysis includes completeness (Comp.), correctness (Corr.), quality (Q.), RMS of boundary error, and height differences. By counting the numbers of correspondences between roof planes in the reference and planes in the reconstruction results, the method uses $N_{1:M}$ as an indicator for over-segmentation, $N_{N:1}$ for under-segmentation, and $N_{N:M}$ indicates clusters of planes that are both over- and under-segmented. The statistics are listed in Table 1. A corresponding illustration is shown in Fig. 13.

Areas 1 and 3 have lower quality scores compared with area 2, because they exhibit some small dormers which are apparently merged into nearby roofs. This implies that small errors caused by dormers or chimneys are difficult to correct automatically. When only considering roofs whose area is larger than 10 m^2 , i.e. containing more than 40 points, all per-roof scores increase. As shown in Fig. 13 (a) and Fig. 13(c), some false positives and false negatives exist at the building boundaries, mainly at step chains. The main reason is that the regularization step can move step contours at boundaries to meet the angle constraints. Fig. 13(b) indicates that the results of roof plane extraction

¹ For interpretation of color in Fig. 11, 14, 15 and 18, the reader is referred to the web version of this article.

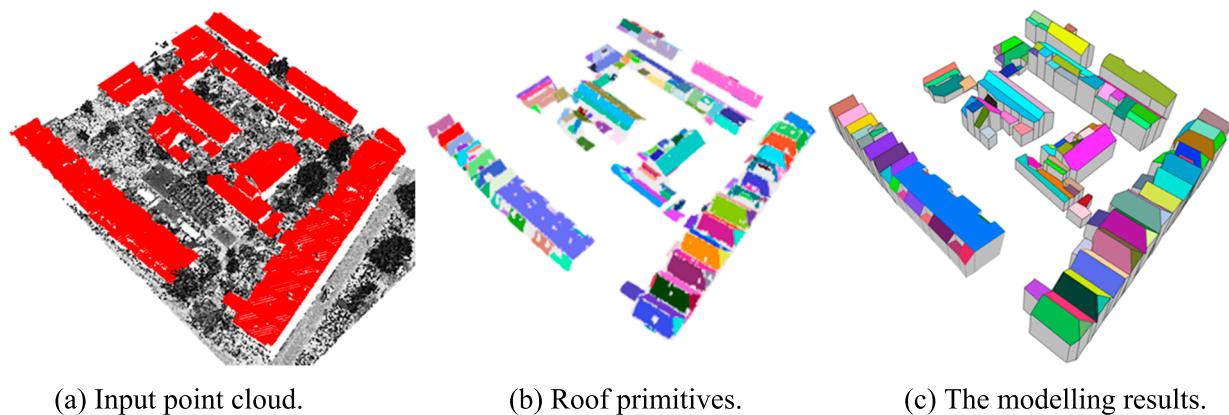


Fig. 11. The urban modelling experiment on the AHN3 dataset.

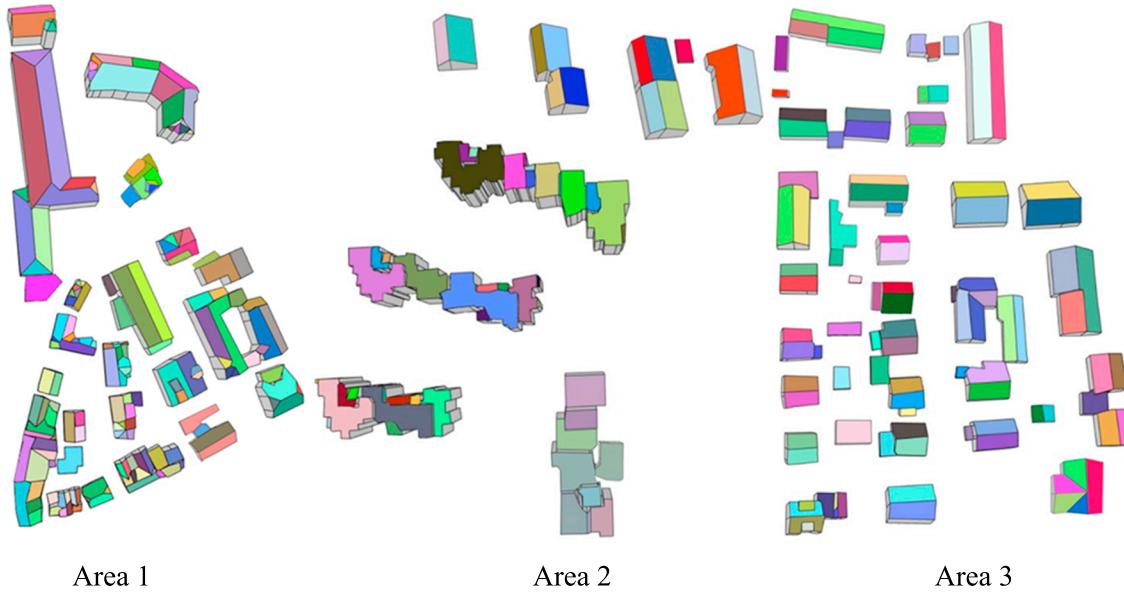


Fig. 12. Modelling results for the Vaihingen dataset.

Table 1
Evaluation results with ISPRS references.

	Per-area (%)			Per-roof (%)			Per-roof Area > 10 m ² (%)			N _{1:M} /N _{N:1} /N _{N:M}	RMS of differences (m)
	Comp.	Corr.	Q.	Comp.	Corr.	Q.	Comp.	Corr.	Q.		
Area 1	87.5	97.9	85.8	83.3	98.1	82.0	87.8	99.2	87.1	1/45/6	0.75
Area 2	91.3	97.0	88.8	91.3	97.0	88.8	95.8	97.7	93.7	2/7/0	0.80
Area 3	86.8	97.5	84.9	79.6	100.0	79.6	91.9	100.0	91.9	1/51/0	0.94

are in good agreement with the reference data. Only very few edges are exceeding or insufficient to cover the reference roofs. The model results show a good performance in retaining the roof plane positions, so the height errors of the roofs are small.

4.3. Comparisons

Extracting roof primitives is crucial in modelling buildings, as mentioned by Lafarge and Mallet (2012) and Chen et al. (2017), because these primitives decide the completeness of the models. RANSAC based roof primitive extraction methods (Schnabel et al., 2007; Henn et al., 2013) are very popular for roof extraction, however such methods are sensitive to non-uniform point density, especially for point clouds obtained from a long distance. A comparison with RANSAC

based method (Schnabel et al., 2007) is shown in Fig. 14, where the RANSAC method uses a fitting threshold of 0.1 m and a minimum supporting point number of 40. In our method, the minimum supporting area A is 1 m². RANSAC suffers from under-segmentation when there are not enough supporting points for the detection, while selecting a small supporting threshold causes over-segmentation. As the parts indicated by the red arrows in Fig. 14(b) show, some small roof planes were missed. Fig. 14(c) demonstrates the completeness of our roof extraction results. Our approach has a similar performance to the RANSAC-based method in terms of fitting error and computational time. A quantitative comparison is given in Table 2.

Furthermore, we compared the proposed approach with the methods of Lafarge and Mallet (2012) and Zhou and Neuman (2010) on data sources these authors have released. Fig. 15(a) shows the input

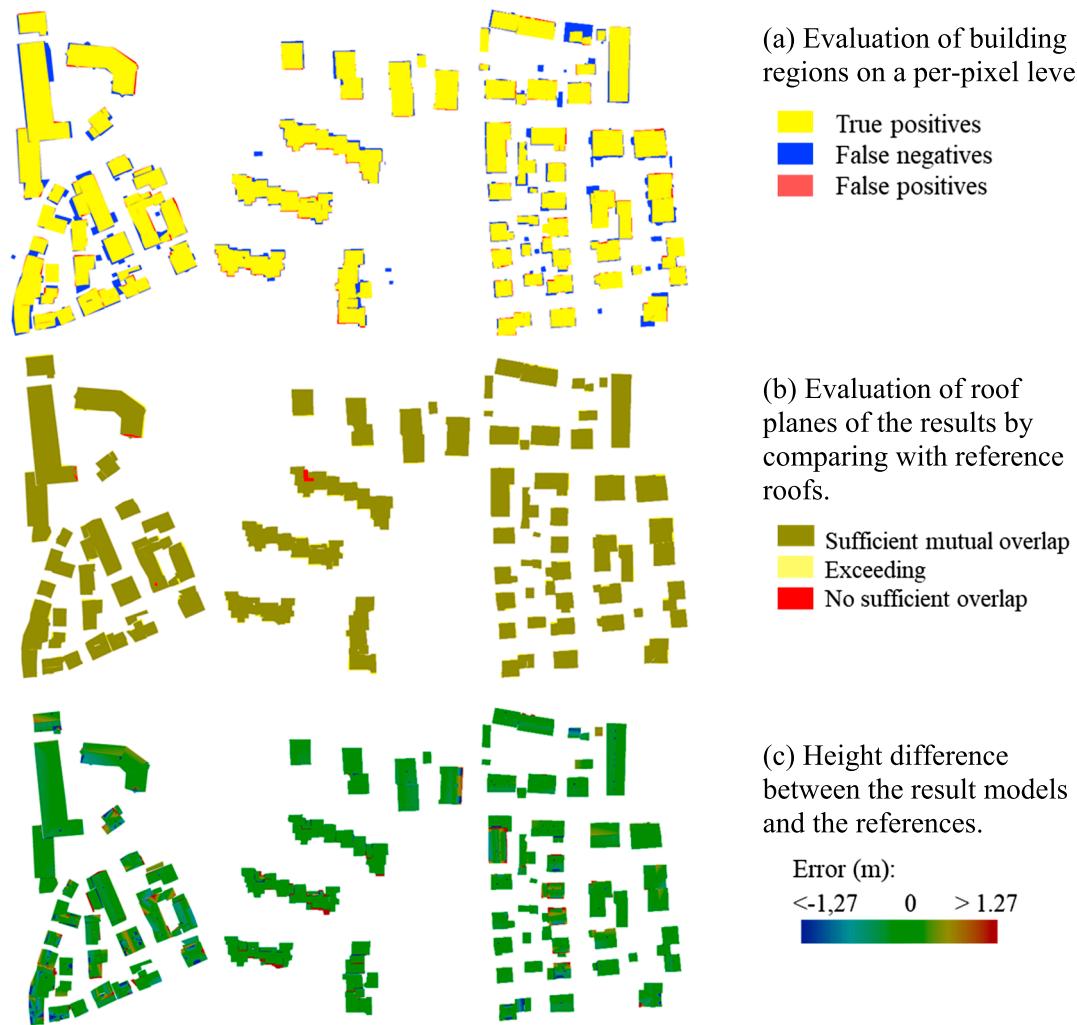


Fig. 13. Evaluation of Vaihingen data test based on the benchmark references.

point clouds coloured by height. In Fig. 15(b) and (c), the results generated by the primitive-based method (Lafarge and Mallet, 2012) and 2.5D dual contouring (D-C) (Zhou and Neuman, 2010) are shown.

The results of the primitive-based method by (Lafarge and Mallet, 2012), see Fig. 15(b), is successful in keeping the surface compact, because it uses the detected lines to sharpen the boundaries. However, their line segment detection method contains a host of parameters which need to be chosen empirically and is therefore difficult to transfer from one dataset to another. Besides, if no boundary segments are detected, details are lost, such as the parts indicated by the red arrows.

The 2.5D D-C method (Zhou and Neuman, 2010) is basically a mesh simplification algorithm which uses statistical information of points in

Table 2
Comparison of roof primitive extraction.

Method	No. of Primitives	Average fitting error (m)	Timing (ms) ¹
Manual	35	—	—
RANSAC based method	18	0.045	258
The proposed method	41	0.032	272

¹ The experiments were performed on identical hardware, a laptop with a 2.50 GHz Intel (R) Core (TM) i7-6500U CPU and 8 GB RAM. Obviously, timing also depends on the code quality. So the given time differences should not be over-interpreted.

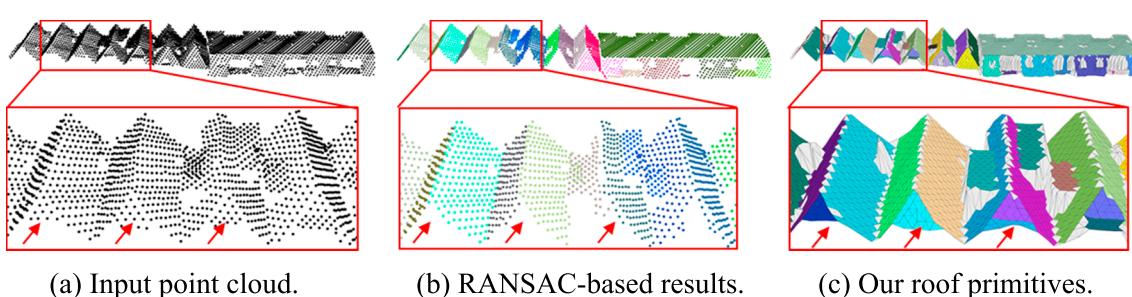


Fig. 14. Comparison of RANSAC-based and the proposed roof primitive extraction.

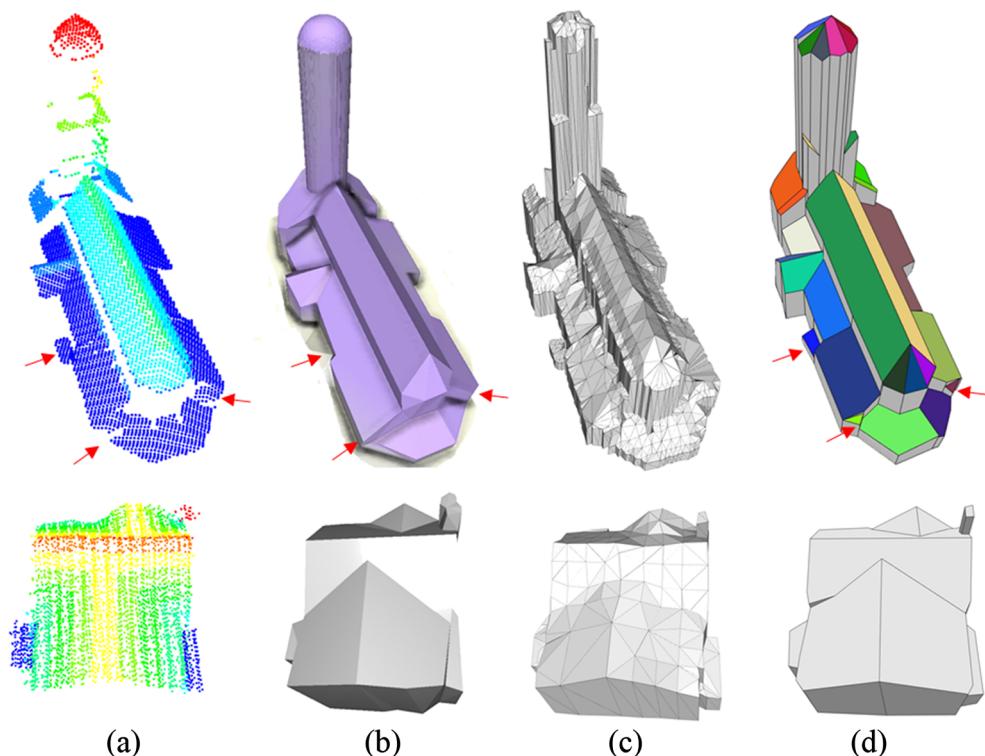


Fig. 15. Comparison of modelling results. (a) Input point cloud coloured by height; (b) modelling results from [Lafarge and Mallet \(2012\)](#); (c) 2.5D D-C modelling results of [Zhou and Neuman \(2010\)](#); and (d) our modelling results.

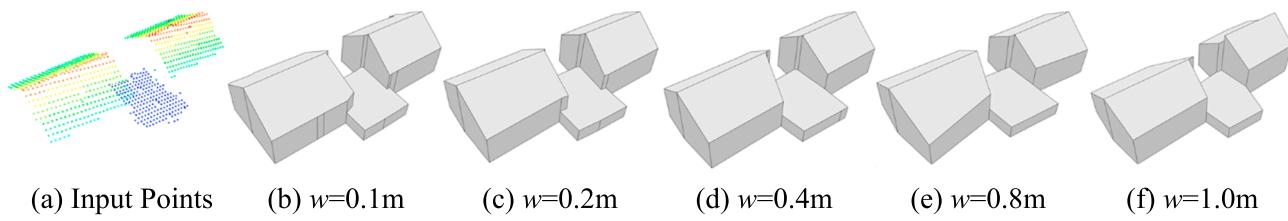


Fig. 16. Modelling results corresponding to different grid widths (w) of the label maps.

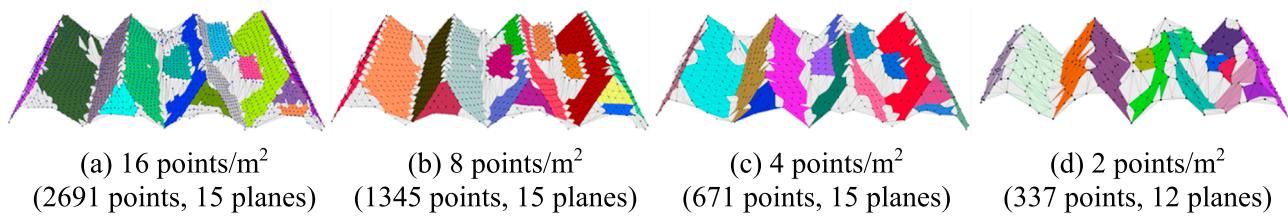


Fig. 17. The adaptive test on various point densities.

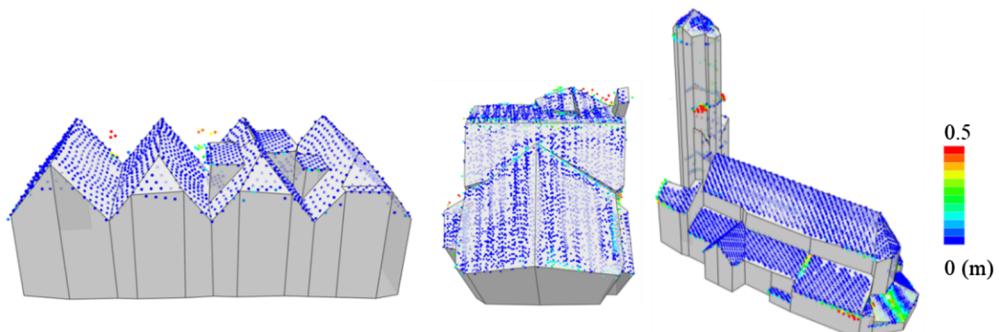


Fig. 18. An illustration of model fitting errors. The colour bar depicts the fitting error magnitude.

Table 3

Statistical information of the experiments for the 3 buildings shown in Fig. 18.

Data in Fig. 18	No. of points	Density/m ²	No. of TIN triangles	No. of primitives	No. of model facets	Timing (ms)	Fitting error RMS (m)
1st col.	1345	8	2596	16	141	301	0.11
2nd col.	4443	17	8800	12	95	236	0.15
3rd col.	5826	2	11,278	32	294	6730	0.22

neighbouring grid cells to smooth the mesh surfaces. It lacks structural detection, so there is no semantic partition across the roofs, see Fig. 15(c). Another problem of the 2.5D D-C method is that the cells of the 2D grid are used only when they contain at least one point, so a small grid size may result in some empty cells while a large grid size may blur the roof edges.

As shown in Fig. 15(d), our final model is compact and we have a similar performance to the primitive-based method of Lafarge and Mallet (2012). Even though the round vault in the first building (the first row) is approximated by a set of small planar patches, the result preserves most details contained in the raw point cloud.

4.4. The effects of parameter selection

Grid width. The grid width controls the model sensitivity. Fig. 16 demonstrates the effects of different grid width values on the modelling results. Smaller widths result in more contour points and vice versa. In our example, as the grid width increases the model tends to be unstable, because if the spatial resolution of the grid is larger than the intersection query threshold Δ , the intersection detection fails. In practice, the grid size is set in the range of 0.5–0.2 m, which works well for the investigated datasets with point densities ranging from 4 points/m² to 10 points/m².

In this paper, the grid orientation was chosen to be parallel to the axes direction of original coordinate framework. While this choice may have an effect on the results in principle, in preliminary experiments we did not observe any visible differences in our results when varying the grid direction.

Point density. The TIN-based primitive detection method can handle different point densities, because in the TIN each triangle has clearly defined neighbours. In addition, every triangle facet implies a local surface direction, which avoids having to estimate less stable point normals. As shown in Fig. 17, even though the point density has been changed from 16 points/m² to 2 points/m², the roof detection algorithm still extracts the main roof structures.

Fitting error. Because ground truth is often lacking, a frequently used method for quality assessment of model results is to compute the RMS of the surface fitting errors which denote the perpendicular distances from points to their corresponding nearest facets in the model (Lafarge and Mallet, 2012; Poullis, 2013; Li et al., 2015). To intuitively evaluate the accuracy of the reconstructed models, we show the point clouds with error related colours superimposed on the polygonal models, see Fig. 18 where the colour bar depicts the error magnitude. The three models correspond to the data shown in Figs. 10 and 15. The statistical information of these experiments is listed in Table 3. The first and second buildings are two residential houses, and the third building is a relatively large church. The computing time for the church is longer than that for the first two buildings. In these experiments, our method has an average fitting error of less than 0.2 m.

As far as the computing time is concerned, the most time-consuming step is graph cut refinement. According to Boykov and Kolmogorov (2004), the computational complexity of the graph cut algorithm applied to grid maps is in the order of $O(n^2\sqrt{m})$, where n is the number of grid cells and m is the number of edges in the graph. Generally, the balance between model accuracy and computational efficiency thus

largely depends on the width of the supporting grid. A comparison of different grid width settings is given in Fig. 16.

In summary, our method creates watertight models for complicated buildings using aerial LiDAR point clouds, and the models contain only a limited number of polygonal facets, which is advantageous for storage, web transmission, and fast rendering. Nevertheless, the approach still has a few limitations. First, the error of the input point cloud is required to be less than the average point spacing, otherwise the TIN model may show artefacts. Another limitation is that we use a piecewise plane assumption, so curved roofs will be approximated by planar patches. Furthermore, for small roof structures with limited point numbers, the approach may fail to recover its original contour. Besides, LiDAR data typically contain gaps, which may result from a gap between objects, e.g. a small alley in between two buildings, or from missing data, e.g. caused by absorption of the laser energy on asphalt or specular reflection on wet roof surfaces. Detecting the nature of these gaps by data-driven methods is a difficult problem. A possible solution for classifying these gaps into one of the mentioned classes is to combine the laser data with imagery. However, such solutions are beyond the scope of this paper.

5. Conclusions

In this paper, we present an automatic approach for the generation of building models from aerial LiDAR point clouds based on a sequence of label map operations. The suggested TIN-based roof primitive detection method is tolerant with respect to varying point density, and the label map facilitates the computation of precise roof reconstruction. We introduce a contour vertex refinement algorithm, which uses filtered segment normal vectors to guide vertex updating. This refinement is based on soft angle constraints that provide a compromise between noise tolerance and shape regularization. Quantitative evaluations using different test datasets show that when roof planes contain a minimum of about 40 points, the method can reconstruct compact models. For the cases investigated in this paper, the average fitting error of the modelling results is approximately 0.2 m. Experiments demonstrate that the approach can be successfully applied to aerial LiDAR point clouds under some challenging conditions. In future work, we will strive to incorporate independently fitted pre-selected patterns into the label map to better handle small roof structures.

Acknowledgements

The authors would like to thank all data providers for sharing their data. The Vaihingen data set 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>. This work was supported in part by the National Natural Science Foundation of China under Grant: 41801342 and the Natural Science Foundation of Jiangsu Province, China, under Grant: BK20170781.

Appendix A. Supplementary material

Supplementary data to this article can be found online at <https://doi.org/10.1016/j.isprsjprs.2019.06.003>.

References

- AHN3 Dataset, 2018. Actueel Hoogtebestand Nederland (AHN), <<https://www.pdok.nl/nl/ahn3-downloads>> .
- Albers, B., Kada, M., Wichmann, A., 2016. Automatic extraction and regularization of building outlines from airborne LiDAR point clouds. *Int. Arch. Photogram. Remote Sens. Spatial Inf. Sci. XLI-B3*, 555–560.
- Blomley, R., Jutzi, B., Weinmann, M., 2016. Classification of airborne laser scanning data using geometric multi-scale features and different neighbourhood types. *ISPRS Ann. Photogram. Remote Sens. Spatial Inf. Sci. III-3*, 169–176.
- Boykov, Y., Kolmogorov, V., 2004. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE Trans. Pattern Anal. Mach. Intell.* 26 (9), 1124–1137.
- Chen, D., Wang, R., Peethambaran, J., 2017. Topologically aware building rooftop reconstruction from airborne laser scanning point clouds. *IEEE Trans. Geosci. Remote Sens.* 55, 7032–7052.
- Chen, L., Zhao, S., 2012. Building detection in an urban area using LiDAR data and quickbird imagery. *Int. J. Remote Sens.* 33 (15), 5135–5148.
- Chen, Y., Cheng, L., Li, M., Wang, J., Tong, L., Yang, K., 2014. Multiscale grid method for detection and reconstruction of building roofs from airborne LiDAR data. *IEEE J. Select. Top. Appl. Earth Observ. Remote Sens.* 7 (10), 4081–4094.
- Cramer, M., 2010. The DGPF test on digital aerial camera evaluation – overview and test design. *Photogrammetr. – Fernerkund. – Geoinform.* 2 (2010), 73–82.
- Dorninger, P., Nothegger, C., 2007. 3D segmentation of unstructured point clouds for building modelling. *Int. Arch. Photogram. Remote Sens. Spatial Inf. Sci.* 36 (3/W49A), 191–196.
- Douglas, D.H., Peucker, T.K., 1973. Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. *Cartographica: Int. J. Geogr. Inf. Geovis.* 10 (2), 112–122.
- Fischler, M.A., Bolles, R.C., 1981. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM* 24, 381–395.
- Haala, N., Hastedt, H., Wolf, K., Ressl, C., Baltrusch, S., 2010. Digital photogrammetric camera evaluation generation of digital elevation models. *Photogrammetrie Fernerkundung Geoinformation* 2010 (2), 99–115.
- Henn, A., Gröger, G., Stroh, V., Plümer, L., 2013. Model driven reconstruction of roofs from sparse LiDAR point clouds. *ISPRS J. Photogramm. Remote Sens.* 76, 17–29.
- INRIA dataset, 2012. <https://www.sop.inria.fr/members/Florent.Lafarge/benchmark/3D-object_detection/3D-objects.html> .
- Kolmogorov, V., Zabih, R., 2004. What energy functions can be minimized via graph cuts? *IEEE Pattern Anal. Mach. Intell.* 26 (2), 147–159.
- Lafarge, F., Descombes, X., Zerubia, J., Pierrot-Deseilligny, M., 2010. Structural approach for building reconstruction from a single DSM. *IEEE Pattern Anal. Mach. Intell.* 32 (1), 135–147.
- Lafarge, F., Mallet, C., 2012. Creating large-scale city models from 3D point clouds: a robust approach with hybrid representation. *Int. J. Comput. Vision* 99 (1), 69–85.
- Li, M., Nan, L., Smith, N., Wonka, P., 2015. Reconstructing building mass models from UAV images. *Comput. Graph.* 54 (C), 84–93.
- Li, M., Nan, L., Liu, S., 2016a. Fitting boxes to Manhattan scenes using linear integer programming. *Int. J. Digital Earth* 9 (8), 806–817.
- Li, M., Wonka, P., Nan, L., 2016b. Manhattan-world urban reconstruction from point clouds. In: 2016, Lecture Notes in Computer Science, ECCV, 2016, pp. 54–69.
- Lin, H., Gao, J., Zhou, Y., Lu, G., Ye, M., Zhang, C., Liu, L., Yang, R., 2013. Semantic decomposition and reconstruction of residential scenes from LiDAR data. *ACM Trans. Graphics* 32 (4), 1–10.
- Meng, X., Wang, L., Currit, N., 2009. Morphology-based building detection from airborne LiDAR data. *Photogramm. Eng. Remote Sens.* 75 (4), 437–442.
- Miliaresis, G., Kokkas, N., 2007. Segmentation and object-based classification for the extraction of the building class from LiDAR DEMs. *Comput. Geosci.* 33, 1076–1087.
- Musialski, P., Wonka, P., Aliaga, D.G., Wimmer, M., Gool, L., Purgathofer, W., 2013. A survey of urban reconstruction. *Comput. Graphics Forum* 32, 146–177.
- Nan, L., Wonka, P., 2017. PolyFit: polygonal surface reconstruction from point clouds. In: 2017 IEEE International Conference on Computer Vision (ICCV). Venice, Italy, Oct. 2017.
- Niemeyer, J., Rottensteiner, F., Soergel, U., Heipke, C., 2016. Hierarchical higher order CRF for the classification of airborne LiDAR point clouds in urban areas. *Int. Arch. Photogram. Remote Sens. Spatial Inf. Sci.* 41, 655–662.
- Nurunnabi, A., Belton, D., West, G., 2012. Robust segmentation in laser scanning 3D point cloud data. In: IEEE International Conference on Digital Image Computing Techniques and Applications (DICTA), 2012, pp. 1–8.
- Oude Elberink, S., Vosselman, G., 2009. Building reconstruction by target based graph matching on incomplete laser data: analysis and limitations. *Sensors* 9 (8), 6101–6118.
- Overby, J., Bodum, L., Kjems, E., Iisoe, P., 2004. Automatic 3D building reconstruction from airborne laser scanning and cadastral data using Hough transform. *Int. Arch. Photogram. Remote Sens.* 35 (B3), 296–301.
- Perera, S.N., Nalani, H.A., Schneider, D., 2014. Integration of image data for refining building boundaries derived from point clouds. *Int. Arch. Photogram. Remote Sens. Spatial Inf. Sci. XL-3*, 5–7.
- Poullis, C., 2013. A framework for automatic modeling from point cloud data. *IEEE Trans. Pattern Anal. Mach. Intell.* 35 (11), 2563–2575.
- Poullis, C., You, S., 2009. Automatic reconstruction of cities from remote sensor data. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2009, Miami, FL, USA.
- Rottensteiner, F., 2006. Consistent estimation of building parameters considering geometric regularities by soft constraints. *Int. Arch. Photogram. Remote Sens. Spatial Inf. Sci.* 36 (3), 13–18.
- Rottensteiner, F., Sohn, G., Gerke, M., Wegner, J., Breitkopf, U., Jung, J., 2014. Results of the ISPRS benchmark on urban object detection and 3D building reconstruction. *ISPRS J. Photogramm. Remote Sens.* 93 (7), 256–271.
- Rottensteiner, F., Trinder, J., Clode, S., Kubik, K., 2005. Automated delineation of roof planes from LiDAR data. *Int. Arch. Photogram. Remote Sens. Spatial Inf. Sci.* 36 (3/W19), 221–226.
- Rutzinger, M., Rottensteiner, F., Pfeifer, N., 2009. A comparison of evaluation techniques for building extraction from airborne laser scanning. *IEEE J. Select. Top. Appl. Earth Observ. Remote Sens.* 2 (1), 11–20.
- Sampath, A., Shan, J., 2010. Segmentation and reconstruction of polyhedral building roofs from aerial LiDAR point clouds. *IEEE Trans. Geosci. Remote Sens.* 48, 1554–1567.
- Schnabel, R., Wahl, R., Klein, R., 2007. Efficient RANSAC for point cloud shape detection. *Comput. Graphics Forum* 26 (2), 214–226.
- Serna, A., Marcotegui, B., 2014. Detection, segmentation and classification of 3d urban objects using mathematical morphology and supervised learning. *ISPRS J. Photogramm. Remote Sens.* 93, 243–255.
- Sohn, G., Jwa, Y., Jung, J., Kim, H.B., 2012. An implicit regularization for 3d building rooftop modeling using airborne LiDAR data. *ISPRS Ann. Photogram. Remote Sens. Spatial Inf. Sci.* I-3, XXII, 305–310.
- Sun, S., Salvaggio, C., 2013. Aerial 3D building detection and modeling from airborne LiDAR point clouds. *IEEE J. Select. Top. Appl. Earth Observ. Remote Sens.* 6 (3), 1440–1449.
- Tarsha-Kurdi, F., Landes, T., Grussenmeyer, P., 2007. Hough transform and extended RANSAC algorithms for automatic detection of 3D building roof planes from LiDAR data. *Int. Arch. Photogram. Remote Sens. XXXVI*, 407–412.
- Verdie, Y., Lafarge, F., Alliez, P., 2015. LOD generation for urban scenes. *ACM Trans. Graphics, Association for Computing Machinery* 34 (3), 30.
- Verma, V., Kumar, R., Hsu, S., 2006. 3D building detection and modeling from aerial lidar data. In: Proceedings IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06), Washington, DC, USA, pp. 2213–2220.
- Vo, A.V., Truong-Hong, L., Laefer, D.F., Bertolotto, M., 2015. Octree based region growing for point cloud segmentation. *ISPRS J. Photogramm. Remote Sens.* 104, 88–100.
- Vosselman, G., Gorte, B.G.H., Sithole, G., Rabbani, T., 2004. Recognizing structure in laser scanning point clouds. In: Proceedings of the ISPRS working group VIII/2: laser scanning for forest and landscape assessment, Freiburg, October 3–6, 2004, pp. 33–38.
- Wang, R., Peethambaran, J., Chen, D., 2018. LiDAR point clouds to 3-D urban models: a review. *IEEE J. Select. Top. Appl. Earth Observ. Remote Sens.* 11 (20), 606–627.
- Weinmann, M., Jutzi, B., Hinz, S., Mallet, C., 2015a. Semantic point cloud interpretation based on optimal neighborhoods, relevant features and efficient classifiers. *ISPRS J. Photogramm. Remote Sens.* 105, 286–304.
- Weinmann, M., Schmidt, A., Mallet, C., Hinz, S., Rottensteiner, F., Jutzi, B., 2015b. Contextual classification of point cloud data by exploiting individual 3D neighborhoods. *ISPRS Ann. Photogram. Remote Sens. Spatial Inf. Sci. II-3/W4*, 271–278.
- Wichmann, A., Agoub, A., Kada, M., 2018. RoofN3D: deep learning training data for 3d building reconstruction. *Int. Arch. Photogram. Remote Sens. Spatial Inf. Sci. XLII-2*, 4–7.
- Xiong, B., Elberink, S.O., Vosselman, G., 2014. A graph edit dictionary for correcting errors in roof topology graphs reconstructed from point clouds. *ISPRS J. Photogramm. Remote Sens.* 93, 227–242.
- Xiong, B., Elberink, S.O., Vosselman, G., 2016. Footprint map partitioning using airborne laser scanning data. *ISPRS Ann. Photogram. Remote Sens. Spatial Inf. Sci. III-3*, 241–247.
- Xiong, B., Jancosek, M., Elberink, S.O., Vosselman, G., 2015. Flexible building primitives for 3D building modeling. *ISPRS J. Photogramm. Remote Sens.* 101, 275–290.
- Yang, B., Xu, W., Dong, Z., 2013. Automated extraction of building outlines from airborne laser scanning point clouds. *IEEE Geosci. Remote Sens. Lett.* 10, 1399–1403.
- Zhang, K., Yan, J., Chen, S.C., 2006. Automatic construction of building footprints from airborne LiDAR data. *IEEE Trans. Geosci. Remote Sens.* 44 (9), 2523–2533.
- Zhou, Q.Y., 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, Geographic Information System, Irvine, California, 2008, Article No. 7.
- Zhou, Q.Y., Neuman, U., 2010. 2.5D dual contouring: a robust approach to creating building models from aerial lidar point clouds. In: Proceedings of the 11th European conference on computer vision conference on computer vision: part III, ECCV10, Crete, Greece, 5–11 September.
- Zhou, Q.Y., Neumann, U., 2012. 2.5D building modeling by discovering global regularities. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 16–21 June, Providence, RI, USA, pp. 326–333.