

Article

Roof Plane Segmentation from Airborne LiDAR Data Using Hierarchical Clustering and Boundary Relabeling

Li Li ¹ , Jian Yao ^{1,2,*}, Jingmin Tu ¹, Xinyi Liu ¹, Yinxuan Li ¹ and Lianbo Guo ³

¹ School of Remote Sensing and Information Engineering, Wuhan University, Wuhan 430000, China; li.li@whu.edu.cn (L.L.); jingmin.tu@whu.edu.cn (J.T.); liuxy0319@whu.edu.cn (X.L.); yinxuanli@whu.edu.cn (Y.L.)

² Shenzhen Jimuyida Technology Co., Ltd., Shenzhen 518000, China

³ School of Wuhan National Laboratory for Optoelectronics, Huazhong University of Science and Technology, Wuhan 430000, China; lbguo@hust.edu.cn

* Correspondence: jian.yao@whu.edu.cn

Received: 22 March 2020; Accepted: 22 April 2020; Published: 25 April 2020



Abstract: The roof plane segmentation is one of the key issues for constructing accurate three-dimensional building models from airborne light detection and ranging (LiDAR) data. Region growing is one of the most widely used methods to detect roof planes. It first selects one point or region as a seed, and then iteratively expands to neighboring points. However, region growing has two problems. The first problem is that it is hard to select the robust seed points. The other problem is that it is difficult to detect the accurate boundaries between two roof planes. In this paper, to solve these two problems, we propose a novel approach to segment the roof planes from airborne LiDAR point clouds using hierarchical clustering and boundary relabeling. For the first problem, we first extract the initial set of robust planar patches via an octree-based method, and then apply the hierarchical clustering method to iteratively merge the adjacent planar patches belonging to the same plane until the merging cost exceeds a predefined threshold. These merged planar patches are regarded as the robust seed patches for the next region growing. The coarse roof planes are generated by adding the non-planar points into the seed patches in sequence using region growing. However, the boundaries of coarse roof planes may be inaccurate. To solve this problem, namely, the second problem, we refine the boundaries between adjacent coarse planes by relabeling the boundary points. At last, we can effectively extract high-quality roof planes with smooth and accurate boundaries from airborne LiDAR data. We conducted our experiments on two datasets captured from Vaihingen and Wuhan using Leica ALS50 and Trimble Harrier 68i, respectively. The experimental results show that our proposed approach outperforms several representative approaches in both visual quality and quantitative metrics.

Keywords: roof plane segmentation; hierarchical clustering; boundary relabeling; airborne LiDAR; building reconstruction

1. Introduction

Automatic three-dimensional (3D) building reconstruction in urban areas is an important and classical research topic in the fields of photogrammetry, remote sensing and computer vision. Because the airborne light detection and ranging (LiDAR) data can directly acquire the 3D information of buildings, it becomes a widely used data for 3D building reconstruction. Most of the 3D building reconstruction approaches can be divided into two general categories: model-driven (top-down) and data-driven (bottom-up) [1]. The model-driven approach assumes that a building is comprised of

some roof primitives with predefined topology. This kind of approach is especially appropriate for constructing 3D building models with relatively simple structure from low-density LiDAR points. For the buildings with complex roof structure and high-density point clouds, the data-driven approach is a better choice. The data-driven approach first extracts the main roof primitives from the input point clouds using a point segmentation algorithm, and then analyzes the primitive topology to construct the 3D building models [2–4]. For the polyhedral building model, the planar patch is the basic primitive. Therefore, the high-quality roof planes should be robustly segmented from the input point clouds before the polyhedral building reconstruction. The plane feature is also one of the key features as well as contour and conner features [5–7]. In general, the roof plane segmentation approaches can be divided into three categories: feature clustering-based, model fitting-based, and region growing-based approaches.

Feature clustering-based approaches use some existing clustering techniques, such as mean shift [8], mode-seeking [9], k-plane [10], and fuzzy k-means [11,12], to cluster points into planar patches based on local surface properties. The results of the feature clustering-based approach depend on the quality of surface property calculation, which is strongly influenced by the choice of neighborhood [13]. Kong et al. [9] applied a slope adaptive neighborhood to calculate the point features. The mode-seeking algorithm is applied to extract clusters from the feature space. Sampath and Shan [12] used the Voronoi neighborhood to estimate the surface normals and separated the LiDAR points into planar and nonplanar ones. Then, they applied the fuzzy k-means method to cluster the planar points into roof planes based on their surface normals. Zhang et al. [14] proposed to detect the roof planes from airborne LiDAR data based on spectral clustering of straight-line segments. However, the feature clustering-based approach still suffers from difficulty in neighborhood selection, and is sensitive to the noise and outliers [15].

Model fitting-based approaches can estimate robust planar parameters from the points with high noise and outliers. Hough Transform (HT) [16] and Random Sample Consensus (RANSAC) [17] are two widely employed model fitting methods for roof plane segmentation. The HT has been widely applied to detect planes, cylinders, and spheres from point clouds [18–21]. It first maps each point of input LiDAR data to a parameter space, and the voting accumulators of corresponding shapes are increased. Then, the parameters of corresponding shapes are estimated by searching the local maxima in the parameter space. However, HT is time-consuming, sensitive to the parameter values, and may find the spurious planes [22]. The RANSAC is another widely used method of fitting a model to noisy data [23]. Tarsha-Kurdi et al. [24] compared the HT and RANSAC for building roof plane segmentation from point clouds. They suggested that RANSAC is more efficient than HT in both segmented results and computational time. However, RANSAC is also relatively time-consuming, and cannot handle the problem of spurious planes. Numerous variants have been developed to solve the problems of standard RANSAC [1,25–28]. Xu et al. [25] proposed a weighted RANSAC framework for roof plane segmentation from point clouds. The wighted function considering both the point-to-plane distance and the angular difference is defined to suppress the spurious planes. Li et al. [26] proposed an improved RANSAC method based on Normal Distribution Transformation (NDT) cells to avoid the spurious planes for plane segmentation. The key contribution of this approach is that the planar NDT cell is selected as a minimal sample in each iteration to ensure the correctness of sampling. However, the problem of spurious planes has not been completely resolved yet in the RANSAC-based plane segmentation approaches.

Region growing-based approaches are another popular choice for roof plane segmentation. They first select one point or region as a seed, and then iteratively expand to neighboring points using appropriate criteria. As for the growing criteria, distances of points to planes and angle differences between normal vectors are two widely used similarity measures. Gorte [29] selected a triangle from input Triangulated Irregular Networks (TINs) as a seed, and then applied the region growing to iteratively add the neighboring triangles into the current segment. Zhang et al. [30] proposed to segment the roof plane using the region growing based on a plane-fitting technique. This approach selects the

point with the best planarity as the seed point for region growing. Cao et al. [31] first selected the seed point in a parameter space with reduced dimensions, and then segmented the planar patches in the spatial space using region growing. Xu et al. [32] proposed a voxel-based region growing segmentation method for roof plane extraction. This approach performs region growing in the voxel level instead of the point level [33]. The Robust Principal Component Analysis (RPCA) is applied to estimate the attribute of voxels. To better estimate point normals around the sharp features, Gilani et al. [34] proposed an improved Principal Component Analysis (PCA) method to generate consistent point normals. Region growing-based approaches are easily implemented, and faster than model fitting-based approaches when the point number is large [35]. However, region growing is sensitive to the selection of the seed points or regions, and struggles to detect the accurate boundaries between two smooth regions. In addition, region growing-based approaches may lead to poor segmentation (over- and undersegmentation) for the building with complex structure, especially in the intersection region. To solve these problems, Awrangjeb and Fraser [36] designed several specific principles for different error cases to refine the initial planar segments. Poz and Ywata [28] refined the segments by merging the similar planar segments. Yan et al. [15] proposed a global optimization approach to refine the initial segments generated by a region growing method. This approach can significantly improve the completeness and correctness rates of initial segments and eliminate the cross-planes. However, as it needs to iteratively perform the multi-label optimization to minimize the energy functions, the computational cost of this approach is high.

In this paper, we also propose a region growing-based approach to segment roof planes from airborne LiDAR point clouds. Feng et al. [37] presented an efficient plane extraction approach based on agglomerative hierarchical clustering for organized point clouds. This approach first constructs a graph by dividing the point clouds into regular grids, and then performs hierarchical clustering on this graph. Inspired by this approach, we also apply the hierarchical clustering to iteratively merge initial set of planar patches generated by an octree-based segmentation method. The merged patches are selected as the seeds for region growing. However, there are two differences between our proposed approach with the approach presented in [37]: (1) the airborne LiDAR data used in this paper is unorganized and (2) an octree-based method is applied to generate the initial planar patches. In addition, to solve the problems of oversegmentation, undersegmentation, and spurious planes, we introduce a novel optimization approach based on boundary relabeling [38]. Starting from the plane segments generated by region growing, it continuously refines the planar patches by modifying the boundaries. The workflow of our proposed roof plane segmentation approach is presented in Figure 1. The key contributions of the proposed approach can be summarized as follows.

- We propose a new region growing-based coarse roof plane segmentation approach. It generates the rough planar clusters via an octree-based method, and merges them using a hierarchical clustering method. The merged patches are selected as the robust seeds for region growing.
- We propose a novel boundary relabeling-based roof plane refinement strategy to improve the quality of the initial coarse plane input. We formulate the roof plane refinement as an energy maximization problem and optimize it using boundary relabeling, which is more efficient than the global energy optimization approach [15]. It can remove most of the errors existed in the coarse segmentation and significantly improve the accuracy of the boundaries between adjacent roof planes.

The rest of this paper is organized as follows. Section 2 introduces our proposed region growing-based coarse plane segmentation approach. The refinement of planar patches using boundary relabeling is presented in Section 3. Experimental results on two sets of airborne LiDAR point clouds captured from Vaihingen and Wuhan are presented in Section 4. The conclusions are made in Section 5.

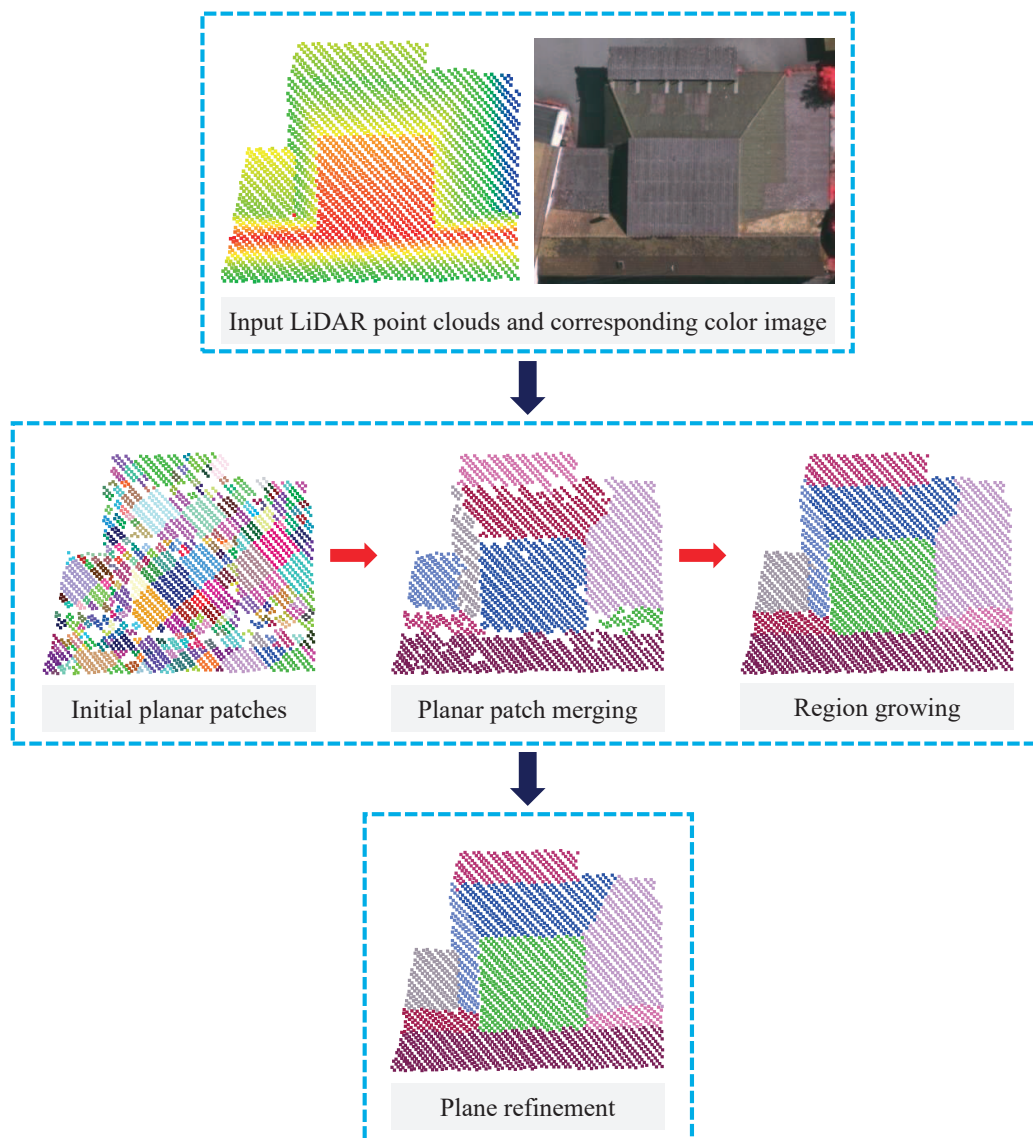


Figure 1. The workflow of our proposed roof plane segmentation approach.

2. Region Growing-Based Coarse Roof Plane Segmentation

In our study, as well as the approaches presented in [15,25], the point clouds of building roofs are first extracted from airborne LiDAR data. To robustly extract roof planes from airborne LiDAR point clouds, we propose a new region growing-based plane segmentation approach. The proposed approach consists of three steps: In the first step, an octree-based segmentation method is proposed to extract the initial planar patches. In the second step, we apply a hierarchical clustering method to iteratively merge adjacent initial patches belonging to the same plane until the merge cost exceeds a predefined threshold. The merged patches are regarded as the seeds for region growing. Finally, the non-planar points are added into the seed patches in sequence via a region growing method. All of the steps of our proposed region growing-based coarse roof plane segmentation approach are presented in Algorithm 1. Moreover, one example of coarse roof plane segmentation is presented in Figure 2.

Algorithm 1 Region growing-based coarse roof plane segmentation**Input:** The input building point clouds \mathbb{R}^3 .**Output:** The coarse roof planes \mathcal{P}_{coarse} .

1. Organize \mathbb{R}^3 into voxels \mathbb{V}^3 .
2. Fit the rough planar patches \mathcal{P}_{rough} via an octree-based segmentation method:
 - (a) For each voxel \mathbf{V}_i , fit the planar patch \mathbf{P}_i ;
 - (b) Find the maximum point-to-plane distance d_{max} among the points included in \mathbf{V}_i ;
 - (c) If $d_{max} \leq T_d$, \mathbf{V}_i is said to be successfully fitted; Otherwise, partition \mathbf{V}_i into eight sub-voxels;
 - (d) Repeat (a)–(c) until each voxel has been successfully fitted or the voxel reaches a minimum size.
3. Merge the rough patches \mathcal{P}_{rough} via hierarchical clustering:
 - (a) Construct the adjacent graph $G = (V, E)$ for the rough patches;
 - (b) Find the edge with the minimum merging MSE, and merge the corresponding patches;
 - (c) Update the adjacent graph;
 - (d) Repeat (b)–(c) until the minimum merging MSE exceeds T_m , obtain the merged patches \mathcal{P}_{merge} .
4. Add the non-planar points into the merged patches \mathcal{P}_{merge} via region growing:
 - (a) Sort \mathcal{P}_{merge} according to the point number they included in a descending order;
 - (b) Iteratively select the seed patch in sequence and grow to its maximum extend.
 - (c) Apply the hierarchical clustering to merge the neighboring patches again, and obtain the coarse roof planes \mathcal{P}_{coarse} .

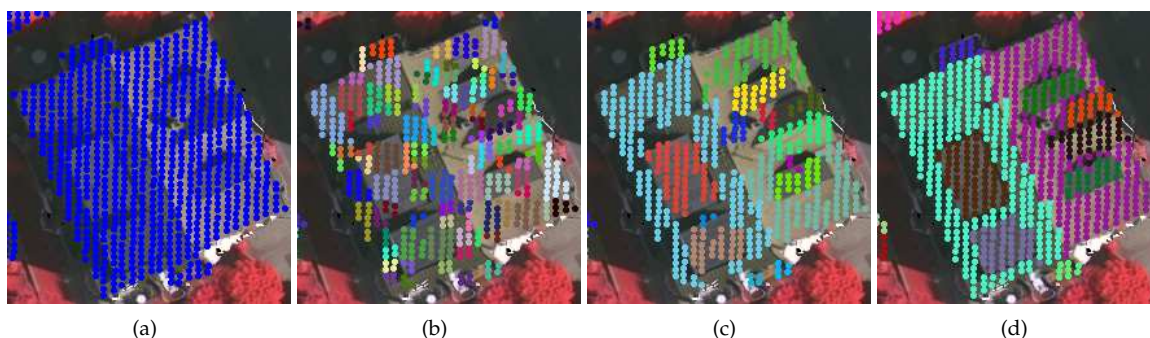


Figure 2. One example of coarse roof plane segmentation. (a) Input light detection and ranging (LiDAR) points (blue points); (b) Octree-based rough planar patch extraction; (c) Rough patch merging; (d) Point-based region growing. The planar patches are denoted by different colors.

2.1. Octree-Based Rough Planar Patch Extraction

Let \mathbb{R}^3 denote the input points. To speed up the plane detection process and to organize the input points, we partition the original data \mathbb{R}^3 into many voxels. Every point belongs to a voxel, and a voxel contains a set of points. Let \mathbb{V}^3 denote all voxels. We apply an octree-based point segmentation approach to efficiently and effectively segment the input points into a set of rough planar patches \mathcal{P}_{rough} . Due to the efficiency of the octree structure, it has been widely used in the field of point cloud segmentation [13,39,40]. For each voxel \mathbf{V}_i , we fit a planar patch \mathbf{P}_i using the PCA. To decide whether this voxel can be regarded as a valid planar patch, we calculate the point-to-plane distance d for each point \mathbf{x} included in \mathbf{V}_i based on the estimated plane parameters. We can easily find the maximal point-to-plane distance d_{max} among all points included in this voxel. If d_{max} is smaller than a predefined threshold T_d , this voxel is said to be successfully fitted; otherwise, this voxel will be continuously divided into eight sub-voxels, and the plane parameters will be estimated for each sub-voxel again. We recursively subdivide the voxel until each sub-voxel can be successfully fitted as a plane or the sub-voxel reaches a minimum size. In general, the size of roof structure is larger than

1 m², so we directly set the minimum voxel size as 1 m. If the sub-voxel reaches a minimum size and still cannot be successfully fitted, the points belonging to this sub-voxel will be regarded as non-planar points. Let \mathcal{P}_{rough} denote all rough planar patches. One visual example of octree-based rough planar patch extraction is shown in Figure 2b.

2.2. Planar Patch Merging Using Hierarchical Clustering

After rough planar patch extraction, a roof may be separated into several small patches, as shown in Figure 2b. Thus, we need an efficient scheme to further merge the patches belonging to the same plane. First, we construct an adjacent graph for the rough planar patches. Let $G = (V, E)$ denote the graph, where V and E denote the nodes and edges, respectively. Each node v of the graph G is a rough patch. Each edge $e_{(p,q)}$ is applied to link two adjacent rough patches $\langle \mathbf{P}_p, \mathbf{P}_q \rangle$. Then, the hierarchical clustering is performed on this graph to iteratively merge rough patches [37,41]. We repeat finding the edge $e_{(p,q)}$ with the minimum merging Mean Squared Error (MSE), and merging two corresponding nodes into one node. The graph is updated after each iteration. The merging MSE is the plane fitting MSE of the union of corresponding two patches $\langle \mathbf{P}_p, \mathbf{P}_q \rangle$. If the minimum merging MSE exceeds the predefined threshold T_m , we terminate the hierarchical clustering and finish the patch merging. Let \mathcal{P}_{merge} denote all merged patches. One visual example of rough planar patch merging is shown in Figure 2c.

2.3. Point-Based Region Growing

After merging rough planar patches, we add the non-planar points into their neighboring patches via region growing to extract the complete roofs. The kd-tree is applied to search the k-nearest neighboring points for each point. We first sort the merged patches \mathcal{P}_{merge} according to the number of points they included in a descending order. The sorted patches are regarded as the seed patches. Then, the point-based region growing [42] is performed to add the non-planar points into the seed patches in sequence. Finally, to avoid the oversegmentation artifacts, the hierarchical clustering is applied again to merge the added patches. One example of final coarse roof plane extraction is shown in Figure 2c. Let \mathcal{P}_{coarse} denote all coarse roof planes.

3. Roof Plane Refinement

In Section 2, the coarse roof planes are segmented from the airborne LiDAR point clouds via a region growing-based approach. In most cases, this region growing-based approach performs well and can extract high-quality roof planes. However, for some buildings with complex roof structure, they still suffer from many problems (spurious planes, oversegmentation, undersegmentation, and so on), especially at the regions close to the boundaries of planar patches. Thus, a new refinement approach based on boundary relabeling is proposed to overcome these problems. We formulate the roof plane refinement as an energy maximization problem and optimize it using boundary relabeling. It refines the planar patches by alternatively relabeling the boundary points into the neighboring optimal patch. Figure 3 presents several representative examples of roof plane refinement. The input point clouds and the corresponding color images are presented in Figure 3a,b, respectively. The coarse roof planes with some representative problems highlighted by the yellow ellipses are presented in Figure 3c. The refined roof planes of Figure 3c are presented in Figure 3d. The representative problems existed in the coarse roof planes have been solved well after the roof plane refinement.

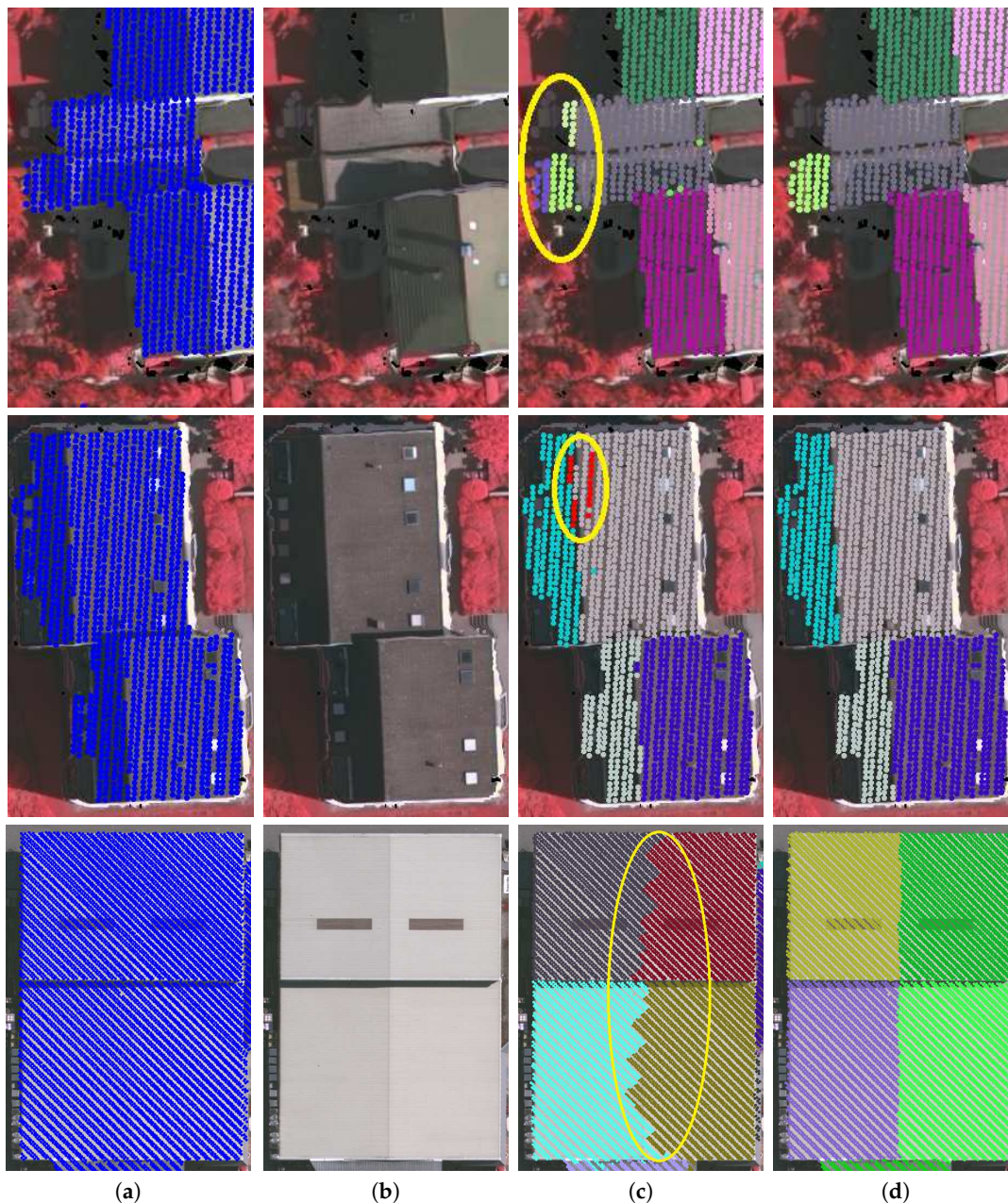


Figure 3. Several representative examples of roof plane refinement. (a) Input airborne LiDAR point clouds; (b) Corresponding color images; (c) Coarse roof planes with the problems of oversegmentation and undersegmentation (the first row), spurious planes (the second row), and irregular boundaries (the third row); (d) Refined roof planes. The yellow ellipses are used to highlight the problems existed in the coarse planes.

3.1. Plane Refinement as an Energy Maximization

Let N be the number of points, and K be the number of roof planes. Let $\mathbf{A} = (A_{x_1}, \dots, A_{x_i}, \dots, A_{x_N})$ be the labels of points. $A_{x_i} = k$ means that the point x_i belongs to the roof plane \mathbf{P}_k . Apparently, we can use \mathbf{A} to represent a partitioning of the points into planar clusters. Let \mathcal{A} denote all valid partitions. The aim of plane refinement is to find the best partition \mathbf{A}^* from \mathcal{A} , namely, assign the best label k to each point x_i . We formulate this problem as an energy optimization problem, and solve it by maximizing the following function,

$$\mathbf{A}^* = \arg \max_{\mathbf{A} \in \mathcal{A}} E(\mathbf{A}), \tag{1}$$

where $E(\mathbf{A})$ is the energy function, which is defined as the sum of distance term $D(\mathbf{A})$ and boundary term $G(\mathbf{A})$. The distance term $D(\mathbf{A})$ is defined based on the point-to-plane distance. The boundary term $G(\mathbf{A})$ is used to ensure that the boundaries of refined roof planes are regular. The energy function is defined as follows,

$$E(\mathbf{A}) = D(\mathbf{A}) + \lambda G(\mathbf{A}) \quad (2)$$

where λ is a weight coefficient, which is applied to balance the influence of each term. The default value of λ is empirically set as 5 according to our testing experience, both the accuracy and regularity of the roof planes can be satisfied. During the testing, we find that λ is insensitive to the input data. λ can be regarded as a constant value in our approach. We simply fix λ as default value in the evaluation experiments of this paper and the results are quite competitive and stable.

3.1.1. Distance Term

The distance term $D(\mathbf{A})$ encourages to assign the point to the plane with the minimum point-to-plane distance. Namely, it attempts to reduce the MSE of the refined roof planes. For each point \mathbf{x}_i and corresponding plane \mathbf{P}_k , we define the local distance term $d(\mathbf{x}_i)$ of \mathbf{x}_i as

$$d(\mathbf{x}_i) = -d(\mathbf{x}_i, \mathbf{P}_k) \quad (3)$$

where $d(\mathbf{x}_i, \mathbf{P}_k)$ is the distance from \mathbf{x}_i to \mathbf{P}_k . Moreover, we define the distance term as

$$D(\mathbf{A}) = \sum_{i=1}^N d(\mathbf{x}_i) \quad (4)$$

Apparently, the distance term achieves the maximum when every point belongs to the plane with the minimum point-to-plane distance. Thus, this term can penalize the appearance of segmentation errors. However, in some cases, the optimal plane for a point is not the plane with minimum point-to-plane distance. One example is presented in Figure 4. From Figure 4a, we can find that there are two planes (Planes A and B) in this building. When only the distance term is used, the refined planes are presented in Figure 4b. In Figure 4b, there are two points marked by the red rectangles. These two points are more closer to the Plane B, but they actually belong to Plane A. Thus, we need to design a boundary term to solve this problem. The refined planes with the use of boundary term are presented in Figure 4c. These two points are assigned to Plane A as expected.

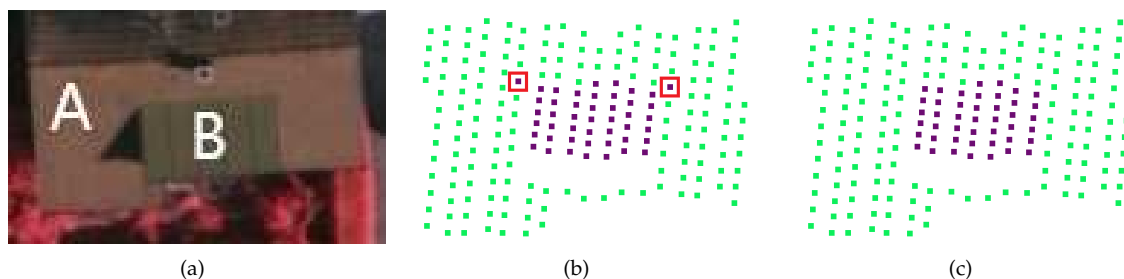


Figure 4. Visual comparison of the roof plane refinement results when the boundary term is or is not used. (a) Corresponding color image; (b) The refined roof planes without the use of boundary term, the points highlighted by the red rectangles are classified into incorrect planar patches; (c) The refined roof planes with the use of boundary term.

3.1.2. Boundary Term

The boundary term $G(\mathbf{A})$ evaluates the regularity of roof planes. It penalizes local irregularities in the boundaries of roof planes, and prefers compactness and smooth boundaries. For each point \mathbf{x}_i , we

assume that its label A_{x_i} equals k . Let \mathcal{N}_i denote the neighboring points of x_i . Let $|\mathcal{N}_i|$ denote the number of points included in \mathcal{N}_i . We define the local smoothness term $g(x_i)$ of point x_i in the area of \mathcal{N}_i as

$$g(x_i) = \frac{1}{|\mathcal{N}_i|} \sum_{j=1}^{|\mathcal{N}_i|} \eta(A_{x_j}, k) \quad (5)$$

where the coefficient $\eta(A_{x_j}, k) = 1$ if $A_{x_j} = k$; otherwise, $\eta(A_{x_j}, k) = 0$. The range of $g(x_i)$ is $[0, 1]$. Based on the local smoothness term, we define the boundary term as

$$G(\mathbf{A}) = \sum_{i=1}^N g(x_i) \quad (6)$$

where N is the total number of points. Apparently, if \mathcal{N}_i contains a unique planar patch, $g(x_i)$ is at its maximum. Near the boundaries, the points included in \mathcal{N}_i belong to several roof planes. Thus, it is not possible that $g(x_i)$ can achieve the maximum in all points. If the number of points close to the boundaries decreases, the value of $G(\mathbf{A})$ will increase. Thus, this term can penalize the jagged boundaries, and can enforce smooth boundaries. In addition, this term can also penalize the island planes or the spurious planes, as these planes will cause longer boundaries.

3.2. Energy Optimization via Boundary Relabeling

We introduce a new optimization method based on boundary relabeling for plane refinement. The coarse roof plane segmentation generated by region growing serves as a good initial partitioning for the next energy optimization. Starting from the initial partitioning \mathbf{A} , it iteratively updates the partitioning by proposing small local change at the boundary regions. Let \mathbf{A}' denote the new partitioning generated by moving a point from one plane to its neighbors. If the energy function $E(\mathbf{A}')$ increases after the local change, the partitioning is updated.

In each iteration, we move only one boundary point to its neighboring planes. Let x_i^b denote a boundary point. We assume that x_i^b belongs to plane \mathbf{P}_p , and $A_{x_i^b} = p$. We propose to move x_i^b from \mathbf{P}_p to its neighboring plane \mathbf{P}_q . We need to efficiently evaluate whether this move is accepted or not. Namely, we need to evaluate whether $E(\mathbf{A}')$ is larger than $E(\mathbf{A})$ or not. Because this move is a local change, it can be efficiently evaluated in a small local region. Before moving x_i^b , we first calculate the local distance term and boundary term, respectively. The local distance term $d(x_i)$ can be easily calculated according to the definition presented in Equation (3). Because the local boundary term $g(x_i)$ is defined around the neighboring points of x_i , the move of x_i will influence the boundary terms of its neighboring points as well as itself. Thus, we calculate the boundary term $g_{\mathcal{N}_i}(x_i)$ for x_i and its neighboring points \mathcal{N}_i as

$$g_{\mathcal{N}_i}(x_i) = \sum_{j=1}^{|\mathcal{N}_i|} g(x_j) \quad (7)$$

where $g(x_j)$ is the local boundary term of a neighboring point x_j and is defined in Equation (5). After moving x_i^b , we calculate the local distance term $d'(x_i)$ and boundary term $g'_{\mathcal{N}_i}(x_i)$ as the similar way. We accept this move if it satisfies

$$\frac{d'(x_i) - d(x_i)}{\max(|d'(x_i)|, |d(x_i)|)} + \lambda \frac{g'_{\mathcal{N}_i}(x_i) - g_{\mathcal{N}_i}(x_i)}{\max(g'_{\mathcal{N}_i}(x_i), g_{\mathcal{N}_i}(x_i))} > 0 \quad (8)$$

If this move has been accepted, we update the partitioning $\mathbf{A} = \mathbf{A}'$. Then, we select the next boundary point, and evaluate this point can be moved or not. The iteration is terminated until no boundary point can be found for relabeling, and the optimal partitioning \mathbf{A}^* is found. Let $\mathcal{P}_{refined}$ denote the final refined roof planes. One example of roof plane refinement via boundary relabeling is presented in Figure 5.

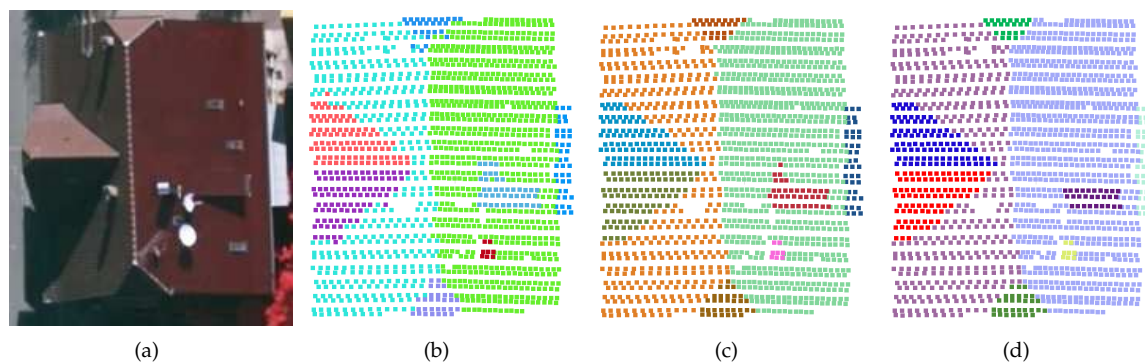


Figure 5. A visual example of roof plane refinement via boundary relabeling. (a) Corresponding color image; (b) Coarse roof planes (initial partitioning); (c) The intermediate refined roof planes with 34 points have been moved; (d) The final refined roof planes with 50 points have been moved.

4. Experimental Results and Discussion

Two datasets are applied to evaluate the proposed approach in the experiments. The first dataset is captured from the city of Vaihingen, Germany [43], provided by the International Society for Photogrammetry and Remote Sensing (ISPRS) (Available at <http://www2.isprs.org/commissions/comm3/wg4/tests.html>). The Vaihingen dataset is captured by a Leica ALS50 system with a mean flying height above ground of 500 m. The maximum viewing angle, pulse repetition rate, and scan rate of this scanner are 37.5 degrees, 83 kHz, and 70 kHz, respectively. The other dataset is captured from the city of Wuhan, China, using a Trimble Harrier 68i sensor. The maximum viewing angle, pulse repetition rate, and scan rate of this scanner are 60 degrees, 400 kHz, and 200 kHz, respectively. The details of two sets of point clouds used in our experiments are presented in Table 1. Several representative local areas of the two datasets are selected to conduct the experiments, as shown in Figures 6 and 7. For these selected areas, the ground truth is generated manually. The ground truth is shown in the second rows of Figures 6 and 7.

Based on these selected areas and corresponding ground truth, we compared the proposed approach with a region growing-based (RG-based) approach [42], a RANSAC-based approach [44], and a global optimization-based (GO-based) approach [15]. The RANSAC-based approach has been implemented in the CloudCompare (Available at <http://www.cloudcompare.org/>) software. In this section, we applied the boundary relabeling-based (BR-based) approach to denote our proposed approach. The proposed approach was implemented with C++ under Windows and tested on a computer with an Intel Core i7-8700 at 3.2 GHz and 16 GB RAM memory.

Table 1. The details of two datasets used in our experiments.

	Vaihingen Dataset	Wuhan Dataset
Location	Vaihingen (Germany)	Wuhan (China)
Laser scanner	Leica ALS50	Trimble Harrier 68i
Point density	4 points/m ²	8 points/m ²
Roof type	Mostly gable roof with a large slope	Flat roof and gable roof
Description of the study areas	Small-sized and detached buildings with complex roof structure	Large-sized buildings with complex roof structure

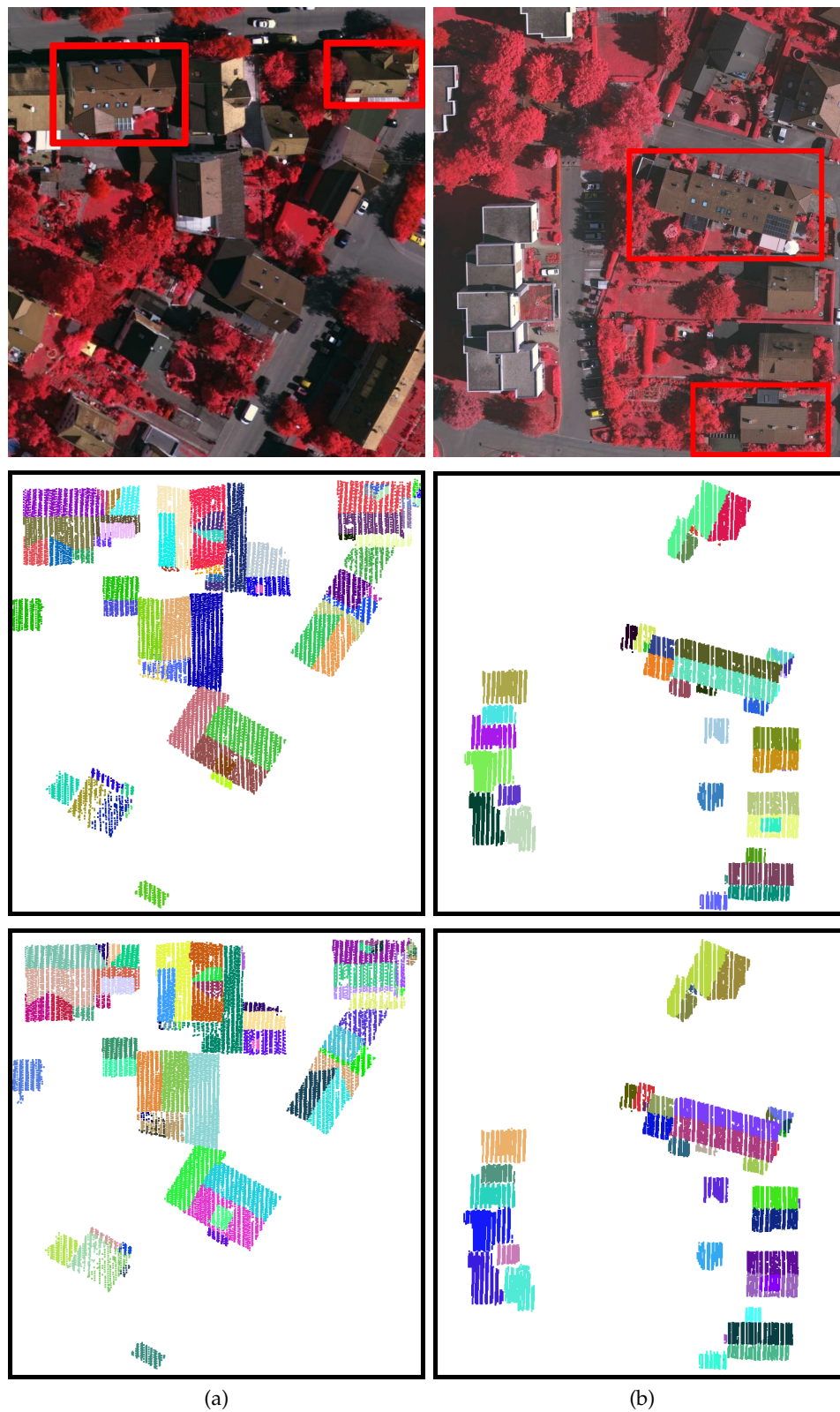


Figure 6. Roof plane segmentation results of Vaihingen dataset. (a,b) are two selected areas. The color images of two areas are presented in the first row. The corresponding ground truth and the segmentation results of the proposed approach are presented in the second and third rows, respectively. The four local buildings highlighted by the red rectangles are selected to conduct the next comparative experiments.

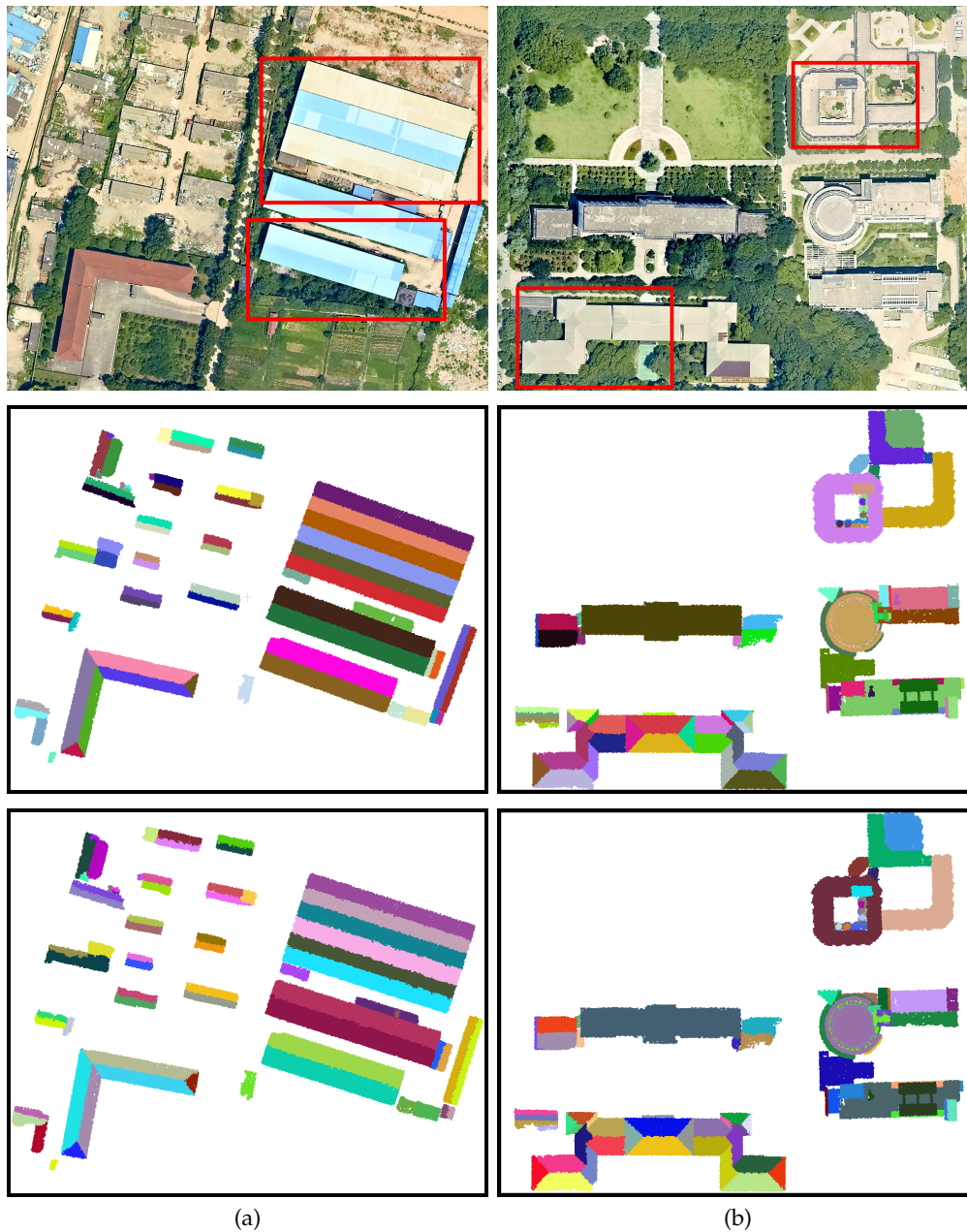


Figure 7. Roof plane segmentation results of Wuhan dataset. (a,b) are two selected areas. The color images of two areas are presented in the first row. The corresponding ground truth and the segmentation results of the proposed approach are presented in the second and third rows, respectively. The four local buildings highlighted by the red rectangles are selected to conduct the next comparative experiments.

4.1. Evaluation Metrics

Several metrics presented in [36] are selected to quantitatively evaluate the performance of the proposed approach. The metrics consist of completeness (C_m), correctness (C_r), and quality (Q_l), which are defined as follows,

$$\begin{aligned}
 C_m &= \frac{TP}{TP + FN} \\
 C_r &= \frac{TP}{TP + FP} \\
 Q_l &= \frac{TP}{TP + FN + FP}
 \end{aligned} \tag{9}$$

where TP (True Positive) is the number of roofs found both in the reference (ground truth) and segmentation, FN (False Negative) is the number of reference roofs not found in segmentation, FP (False Positive) is the number of roofs in segmentation not found in reference. The reference of the datasets are generated manually. Let GT denote the ground truth. To be a true positive, a minimum overlap 50% with the reference is required. In addition, the maximum overlap [25,26] is applied to find the correspondences between the reference and the segmented planes. Only the one-to-one correspondences are established in this method.

To evaluate the errors of oversegmentation and undersegmentation, two additional metrics—the reference cross-lap rate (R_c) and detection cross-lap rate (D_c) [45]—are applied. The reference cross-lap rate is the percentage of the reference planes that overlap multiple detected planes, and the detection cross-lap rate is the percentage of the detected planes that overlap more than one reference plane. Therefore, they can be applied to show the oversegmentation and undersegmentation errors, and they are defined as follows,

$$\begin{aligned} R_c &= \frac{N_r'}{N_r} \\ D_c &= \frac{N_d'}{N_d} \end{aligned} \quad (10)$$

where N_r and N_d denote the number of reference planes and detected planes, respectively. N_r' is the number of reference planes that overlap more than one detected plane. N_d' denotes the number of detected planes that overlap more than one reference plane.

In addition, the boundary precision (B_p), boundary recall (B_r), and average precision-recall known as F-measure (F_m) [46,47] are applied to evaluate the quality of boundaries, and they are defined as follows,

$$\begin{aligned} B_p &= \frac{|C_d \cap C_r|}{|C_d|} \\ B_r &= \frac{|C_d \cap C_r|}{|C_r|} \\ F_m &= \frac{2}{1/B_r + 1/B_p} \end{aligned} \quad (11)$$

where C_d and C_r denote the boundary points of the reference and detected planes, respectively. $||$ denotes the number of points included in a set.

4.2. Choice of Parameters

Two main parameters (T_d and T_m) of the proposed BR-based approach need to be set. In the proposed approach, T_d is applied to control the generation of the rough planar patches and T_m is used to control the merging of neighboring planar patches. In this section, we selected two local buildings from Vaihingen dataset to illustrate how these parameters influence the proposed approach.

In Figure 8, we illustrated the influence of T_d , and the value of T_m is fixed as 0.01. Figure 8a,e presents the input point clouds and corresponding color image, respectively. As T_d decreases, the proposed approach may fail to fit the rough patches for some main roofs. As shown in Figure 8b,f, two roof planes highlighted by the yellow ellipses have been missed. As T_d increases, the accuracy of the rough planar patches will decrease, and the undersegmentation errors may appear. As shown in Figure 8d,h, two similar adjacent roof planes have been regarded as one roof plane. When $T_d = 0.1$ m, the generated rough planar patches and the final roof planes are the best for this building, as shown in Figure 8c,g. We suggested to tune T_d in the range of [0.1 m, 0.2 m] for different datasets according to their point accuracy. We set $T_d = 0.1$ m for both Vaihingen and Wuhan datasets.

To illustrate the influence of T_m , we fixed the value of T_d as 0.1 m, and changed the value of T_m . The influence of T_m is illustrated in Figure 9. Figure 9a presents the test building. As T_m decreases,

the proposed approach may fail to merge the patches belonging to the same roof plane, and the oversegmentation errors may appear, as shown in Figure 9b. As T_m increases, more and more patches will be merged into one patch, and the undersegmentation errors may appear, as shown in Figure 9d. Figure 9c presents the roofs generated using $T_m = 0.01$, and we found that both the over- and undersegmentation errors disappear. The proposed approach is sensitive to the value of T_m , and we suggested to tune T_m in the range of $[0.01, 0.03]$ for different datasets according to their point accuracy. For Vaihingen and Wuhan datasets, we set $T_m = 0.01$ and 0.02 , respectively.

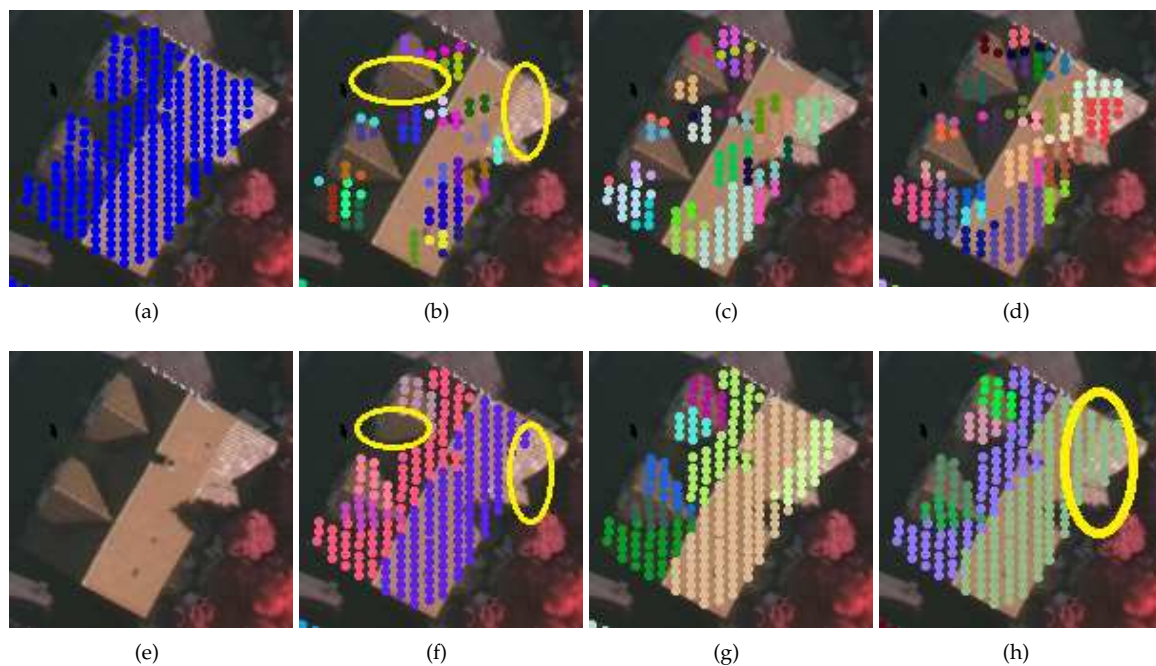


Figure 8. The illustration of T_d . Panels (a,e) are the input LiDAR points and corresponding color image. Panels (b–d) are the rough planar extraction results generated with the parameter T_d as 0.01 m, 0.1 m and 0.2 m, respectively. Panels (f–h) are the corresponding final roof planes of panels (b–d). The yellow ellipses are used to highlight the errors caused by the unsuitable T_d .

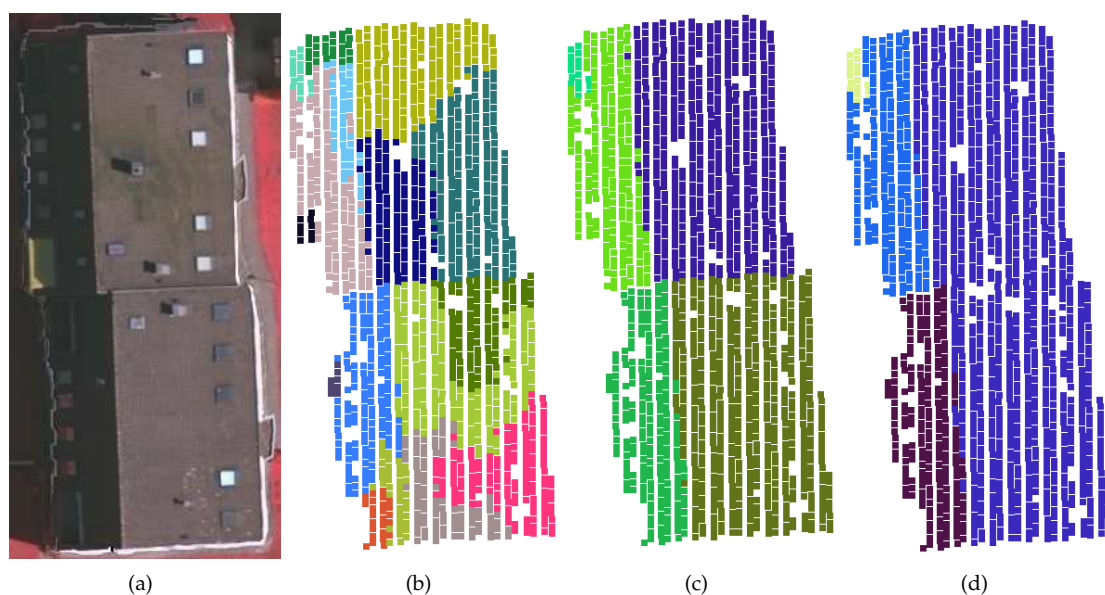


Figure 9. The roof plane segmentation results with different values of T_m . (a) Corresponding color image; (b) $T_m = 0.002$; (c) $T_m = 0.01$; (d) $T_m = 0.02$.

4.3. Comparative Evaluation

We conducted the visual comparison for different approaches on several local buildings selected from Vaihingen and Wuhan datasets, as shown in Figures 10 and 11. In addition, based on the roof planes generated by the proposed BR-based approach, we produced the 3D building models using the method presented in [4]. The corresponding models are presented in the last rows of Figures 10 and 11.



Figure 10. Roof plane segmentation results of different approaches on four local buildings (a–d) selected from the Vaihingen dataset. From the first row to the last row: reference images, ground truth, RG, RANSAC, GO, BR, and 3D models. The black rectangles denote the over- or undersegmentation errors. The red rectangles denote the irregular boundaries.



Figure 11. Roof plane segmentation results of different approaches on four local buildings (a–d) selected from the Wuhan dataset. From the first row to the last row: reference images, ground truth, RG, RANSAC, GO, BR, and 3D models. The black rectangles denote the over- or undersegmentation errors and spurious planes. The red rectangles denote the irregular boundaries. The blue lines denote the real boundaries between adjacent roof planes.

Figure 10 presents the roof segmentation results of four local buildings selected from the Vaihingen dataset (marked by the red rectangles in Figure 6). In Figure 10a, the irregular boundaries appear in the roof plane segmentation results generated by the RG-, RANSAC-, and GO-based approaches. The roof planes extracted by the RG- and RANSAC-based approaches also suffer from the problems of oversegmentation and spurious planes. However, the BR-based approach successfully avoids these problems. In Figure 10b, the proposed BR-based approach suffers from the problems of over- and undersegmentation (highlighted by the black rectangles). The main reason is that the normal difference between the corresponding planes is small. Moreover, the RG- and GO-based approaches

also suffer from the similar problems. In addition, the RG- and RANSAC-based approaches also generate the irregular boundaries (highlighted by the red rectangles). In Figure 10c, the proposed BR-based approach offers the best segmentation. However, the over- or undersegmentation errors, and the spurious planes, appear in the roof segmentation results generated by the rest approaches. In Figure 10d, the GO- and BR-based approaches generate the best roof planes. The roof planes generated by the rest approaches all suffer from the problem of undersegmentation.

Figure 11 presents the roof segmentation results of four local buildings selected from the Wuhan dataset (marked by the red rectangles in Figure 7). In Figure 11a, the building has six roof planes, and the edges between these roof planes are weak. The RANSAC-based approach fails to detect any right roof planes for this building. The RG-, GO-, and BR-based approaches successfully detect the six roof planes. For the RG-based approach, the boundaries are smooth and regular, but almost all boundaries mismatch the ground truth. Both the GO- and BR-based approaches generate the roof planes with smooth boundaries. However, we found that the boundaries of the segments generated by the BR-based approach adhere more tightly to the ground truth. In Figure 11b, although the RG- and RANSAC-based approaches generate smooth and regular boundaries, they mismatch the ground truth. The blue lines in Figure 11b denote the real boundaries between two main roof planes. The BR-based approach still generates the best roof planes, and adheres more tightly to the ground truth than the GO-based approach. In Figure 11c, this building has 15 roof planes. Both the RG- and GO-based approaches generate some small spurious planes (black rectangles). In addition, both the RG- and RANSAC-based approaches generate the irregular boundaries (red rectangles). The BR-based approach generates the best roof planes for this building. In Figure 11d, there is a complex building with a multilayer structure. All approaches generate high-quality roof planes for the top layer of this complex building. However, both the RG- and GO-based approaches fail to detect the right planes for the bottom layer of this building (black rectangles). The RANSAC- and BR-based approaches perform well for this complex building.

Except for the local visual comparison, we performed the global quantitative evaluation for all approaches. Based on the ground truth presented in the second rows of Figures 6 and 7, we calculated the evaluation metrics defined in Section 4.1 for all approaches. The quantitative evaluation results are shown in Table 2. For the first area of Vaihingen dataset (Figure 6a), the proposed BR-based approach offers the best correctness and quality, less oversegmentation errors, best boundary precision and recall, and best F-measure. For the completeness value, the BR-based approach is only slightly lower than the GO-based approach. For the detection cross-rate, the GO-based approach performs better. This is because the GO-based approach tends to segment a roof into many planes for this area, so there are many oversegmentation errors, and the undersegmentation errors will decrease. The value of R_c reaches 17.46%, and it is much larger than 6.35% offered by the BR-based approach. For the second area of Vaihingen dataset (Figure 6b), the proposed BR-based approach offers the best scores for all evaluation metrics. In addition, for the two areas of Wuhan dataset (Figure 7a,b), the BR-based approach still offers the best scores for almost all metrics, except the detection cross-rate in the second area. This is also because the GO-based approach tends to over-segment the roofs for the second area of Wuhan dataset. In addition, for the second area of Wuhan dataset (Figure 7b), the completeness, correctness, boundary precision, boundary recall, and F-measure of the proposed BR-based approach are significantly larger than the rest approaches. The GO-based approach offers relatively bad correctness, many oversegmentation errors, low boundary precision and recall, and low F-measure. In conclusion, the proposed BR-based approach performs best among all approaches on both Vaihingen and Wuhan datasets. The GO-based approach performs well on the Vaihingen dataset, but performs bad on the Wuhan dataset.

At last, we discussed the computational time of all approaches, as shown in the last column of Table 2. We found that the computational time of the RG-based approach is the shortest among all approaches. We also found that the computational time of the GO-based approach is the longest, as it needs to iteratively perform the multi-label optimization to minimize the energy functions. Compared with the GO-based approach, the proposed BR-based approach is very efficient. The computational

time of the proposed BR-based approach is significantly shorter than that of the GO-based approach. In addition, for the Wuhan dataset, the computational time of the BR-based approach is longer than that of the RANSAC-based approach. However, for the Vaihingen dataset, the computational time of the BR-based approach is shorter than that of the RANSAC-based approach.

Table 2. The quality assessment on roof plane segmentation results generated by the different approaches in Figures 6 and 7.

Figure 6a	GT	TP	FN	FP	%C _m	%C _r	%Q _l	%R _c	%D _c	%B _p	%B _r	%F _m	Time(s)
RG	63	58	5	15	92.06	79.45	74.36	22.22	5.48	76.07	84.19	79.92	0.17
RANSAC	63	57	6	23	90.48	71.25	66.28	20.63	5.00	75.57	82.76	79.00	0.91
GO	63	61	2	14	96.83	81.33	79.22	17.46	1.33	81.26	92.38	86.46	11.55
BR	63	60	3	6	95.24	90.91	86.96	6.35	3.03	86.50	92.76	89.52	0.52
Figure 6b													
RG	40	39	1	11	97.50	78.00	76.47	12.50	4.00	88.76	88.29	88.52	0.16
RANSAC	40	38	2	6	95.00	86.36	82.61	10.00	4.55	89.92	90.92	90.42	0.93
GO	40	39	1	3	97.50	92.86	90.70	5.00	0.00	93.65	95.53	94.58	9.37
BR	40	40	0	1	100.00	97.56	97.56	2.50	0.00	95.52	96.24	95.88	0.38
Figure 7a													
RG	68	64	4	31	94.12	67.37	64.65	29.41	13.68	48.64	55.56	51.87	1.17
RANSAC	68	58	10	17	85.29	77.33	68.24	32.35	13.33	47.29	53.06	50.01	1.81
GO	68	67	1	4	98.53	94.37	93.06	5.88	2.82	79.09	81.66	80.35	99.86
BR	68	67	1	3	98.53	95.71	94.37	2.94	1.43	86.61	85.42	86.01	3.96
Figure 7b													
RG	86	74	12	83	86.05	47.13	43.69	25.58	3.82	52.63	54.32	53.46	1.65
RANSAC	86	78	8	35	90.70	69.03	64.46	13.95	7.96	68.71	73.94	71.23	2.23
GO	86	79	7	62	91.86	56.03	53.38	19.77	2.13	67.57	78.15	72.48	390.25
BR	86	80	6	1	93.02	98.77	91.95	1.16	4.94	88.76	85.92	87.32	5.21

5. Conclusions

In this paper, we have presented a novel approach for automatically extracting roof planes with smooth and accuracy boundaries from airborne LiDAR data with the use of hierarchical clustering and boundary relabeling. Especially, the proposed boundary relabeling-based plane refinement approach can remove most of the segmentation errors existed in the coarse segmentation, and significantly improve the quality of roof planes. In addition, as the boundary relabeling is performed on a local region, the computational cost of the proposed approach is significantly lower than the GO-based approach. The experimental results also show that the proposed approach can generate high-quality roof planes, and outperforms the RG-based, the RANSAC-based, and the GO-based approaches.

Nevertheless, the proposed approach still needs to be improved. First, two parameters need to be tuned for different dataset in the proposed approach, and the adaptive thresholding methods should be developed. Second, the plane refinement results of boundary relabeling rely on the coarse plane segmentation input, and the boundary relabeling may suffer from the local optimum. The more flexible and global optimization strategies should be developed to solve these problems.

Author Contributions: Conceptualization, L.L. and J.Y.; methodology, L.L.; software, L.L. and X.L.; validation, L.L. and J.T.; formal analysis, J.Y.; investigation, L.L. and Y.L.; data curation, J.T. and Y.L.; writing—original draft preparation, L.L.; writing—review and editing, J.Y. and L.G.; visualization, L.L. and X.L.; supervision, J.Y. and L.G.; project administration, J.Y.; funding acquisition, J.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Key Research and Development Program of China (Project No. 2017YFB1302400).

Acknowledgments: The authors would like to thank the German Society for Photogrammetry, Remote Sensing, and Geoinformation for providing the Vaihingen dataset [43]. The authors would like to thank the editor and four anonymous reviewers for their valuable comments and suggestions to improve the manuscript.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Chen, D.; Zhang, L.; Li, J.; Liu, R. Urban building roof segmentation from airborne LiDAR point clouds. *Int. J. Remote Sens.* **2012**, *33*, 6497–6515. [[CrossRef](#)]
2. Awrangjeb, M.; Gilani, S.A.N.; Siddiqui, F.U. An effective data-driven method for 3-D building roof reconstruction and robust change detection. *Remote Sens.* **2018**, *10*, 1512. [[CrossRef](#)]
3. Liu, X.; Zhang, Y.; Ling, X.; Wan, Y.; Liu, L.; Li, Q. TopoLAP: Topology recovery for building reconstruction by deducing the relationships between linear and planar primitives. *Remote Sens.* **2019**, *11*, 1372. [[CrossRef](#)]
4. Li, M.; Rottensteiner, F.; Heipke, C. Modelling of buildings from aerial LiDAR point clouds using TINs and label maps. *ISPRS J. Photogramm. Remote Sens.* **2019**, *154*, 127–138. [[CrossRef](#)]
5. Kahaki, S.M.M.; Nordin, M.J.; Ashtari, A.H. Contour-based corner detection and classification by using mean projection transform. *Sensors* **2014**, *14*, 4126–4143. [[CrossRef](#)] [[PubMed](#)]
6. Im, J.H.; Im, S.H.; Jee, G.I. Vertical corner feature based precise vehicle localization using 3D LiDAR in urban area. *Sensors* **2016**, *16*, 1268. [[CrossRef](#)]
7. Hackel, T.; Wegner, J.D.; Schindler, K. Contour detection in unstructured 3D point clouds. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 26 June–1 July 2016; pp. 1610–1618.
8. Melzer, T. Non-parametric segmentation of ALS point clouds using mean shift. *J. Appl. Geod.* **2007**, *1*, 159–170. [[CrossRef](#)]
9. Filin, S.; Pfeifer, N. Segmentation of airborne laser scanning data using a slope adaptive neighborhood. *ISPRS J. Photogramm. Remote Sens.* **2006**, *60*, 71–80. [[CrossRef](#)]
10. Kong, D.; Xu, L.; Li, X.; Li, S. K-plane-based classification of airborne LiDAR data for accurate building roof measurement. *IEEE Trans. Instrum. Meas.* **2014**, *63*, 1200–1214. [[CrossRef](#)]
11. Biosca, J.; Lerma, J. Unsupervised robust planar segmentation of terrestrial laser scanner point clouds based on fuzzy clustering methods. *ISPRS J. Photogramm. Remote Sens.* **2008**, *63*, 84–98. [[CrossRef](#)]
12. Sampath, A.; Shan, J. Segmentation and reconstruction of polyhedral building roofs from aerial LiDAR point clouds. *IEEE Trans. Geosci. Remote Sens.* **2009**, *48*, 1554–1567. [[CrossRef](#)]
13. Vo, A.V.; Truong-Hong, L.; Laefer, D.F.; Bertolotto, M. Octree-based region growing for point cloud segmentation. *ISPRS J. Photogramm. Remote Sens.* **2015**, *104*, 88–100. [[CrossRef](#)]
14. Zhang, C.; He, Y.; Fraser, C.S. Spectral clustering of straight-line segments for roof plane extraction from airborne LiDAR point clouds. *IEEE Geosci. Remote Sens. Lett.* **2018**, *15*, 267–271. [[CrossRef](#)]
15. Yan, J.; Shan, J.; Jiang, W. A global optimization approach to roof segmentation from airborne LiDAR point clouds. *ISPRS J. Photogramm. Remote Sens.* **2014**, *94*, 183–193. [[CrossRef](#)]
16. Duda, R.; Hart, P. Use of the Hough transformation to detect lines and curves in pictures. *Commun. ACM* **1972**, *15*, 11–15. [[CrossRef](#)]
17. Fischler, M.A.; Bolles, R.C. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM* **1981**, *24*, 381–395. [[CrossRef](#)]
18. Vosselman, G.; Gorte, B.; Sithole, G.; Rabbani, T. Recognising structure in laser scanner point clouds. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2004**, *46*, 33–38.
19. Borrmann, D.; Elseberg, J.; Lingemann, K.; Nüchter, A. The 3D Hough transform for plane detection in point clouds: A review and a new accumulator design. *3D Res.* **2011**, *2*, 3. [[CrossRef](#)]
20. Hulik, R.; Spanel, M.; Smrz, P.; Materna, Z. Continuous plane detection in point-cloud data based on 3D Hough transform. *J. Vis. Commun. Image Represent.* **2014**, *25*, 86–97. [[CrossRef](#)]
21. Tian, Y.; Song, W.; Chen, L.; Sung, Y.; Kwak, J.; Sun, S. Fast planar detection system using a GPU-based 3D Hough transform for LiDAR point clouds. *Appl. Sci.* **2020**, *10*, 1744. [[CrossRef](#)]
22. Nguyen, A.; Le, B. 3D point cloud segmentation: A survey. In Proceedings of the IEEE Conference on Robotics, Automation and Mechatronics, Manila, Philippines, 12–15 November 2013.
23. Kahaki, S.M.; Wang, S.L.; Stepanyants, A. Accurate registration of in vivo time-lapse images. In Proceedings of the Medical Imaging 2019: Image Processing, San Diego, CA, USA, 16–21 February 2019.

24. Tarsha-Kurdi, F.; Landes, T.; Grussenmeyer, P. Hough-transform and extended RANSAC algorithms for automatic detection of 3D building roof planes from LiDAR data. *Int. Arch. Photogramm. Remote Sens.* **2007**, *66*, 124–132.
25. Xu, B.; Jiang, W.; Shan, J.; Zhang, J.; Li, L. Investigation on the weighted RANSAC approaches for building roof plane segmentation from LiDAR point clouds. *Remote Sens.* **2016**, *8*, 5. [[CrossRef](#)]
26. Li, L.; Yang, F.; Zhu, H.; Li, D.; Li, Y.; Tang, L. An improved RANSAC for 3D point cloud plane segmentation based on normal distribution transformation cells. *Remote Sens.* **2017**, *9*, 433. [[CrossRef](#)]
27. Canaz Sevgen, S.; Karsli, F. An improved RANSAC algorithm for extracting roof planes from airborne LiDAR data. *Photogramm. Rec.* **2019**. [[CrossRef](#)]
28. Dal Poz, A.P.; Yano Ywata, M.S. Adaptive random sample consensus approach for segmentation of building roof in airborne laser scanning point cloud. *Int. J. Remote Sens.* **2020**, *41*, 2047–2061. [[CrossRef](#)]
29. Gorte, B. Segmentation of TIN-structured surface models. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2002**, *34*, 465–469.
30. Zhang, K.; Yan, J.; Chen, S.C. Automatic construction of building footprints from airborne LiDAR data. *IEEE Trans. Geosci. Remote Sens.* **2006**, *44*, 2523–2533. [[CrossRef](#)]
31. Cao, R.; Zhang, Y.; Liu, X.; Zhao, Z. Roof plane extraction from airborne LiDAR point clouds. *Int. J. Remote Sens.* **2017**, *38*, 3684–3703. [[CrossRef](#)]
32. Xu, Y.; Yao, W.; Hoegner, L.; Stilla, U. Segmentation of building roofs from airborne LiDAR point clouds using robust voxel-based region growing. *Remote Sens. Lett.* **2017**, *8*, 1062–1071. [[CrossRef](#)]
33. Deschaud, J.E.; Goulette, F. A fast and accurate plane detection algorithm for large noisy point clouds using filtered normals and voxel growing. In Proceedings of the International Symposium on 3DPVT, Paris, France, 17–20 May 2010.
34. Gilani, S.A.N.; Awrangjeb, M.; Lu, G. Segmentation of airborne point cloud data for automatic building roof extraction. *GISci. Remote Sens.* **2018**, *55*, 63–89. [[CrossRef](#)]
35. Wu, T.; Hu, X.; Ye, L. Fast and accurate plane segmentation of airborne LiDAR point cloud using cross-line elements. *Remote Sens.* **2016**, *8*, 383. [[CrossRef](#)]
36. Awrangjeb, M.; Fraser, C. An automatic and threshold-free performance evaluation system for building extraction techniques from airborne LiDAR data. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2014**, *7*, 4184–4198. [[CrossRef](#)]
37. Feng, C.; Taguchi, Y.; Kamat, V.R. Fast plane extraction in organized point clouds using agglomerative hierarchical clustering. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Hong Kong, China, 31 May–7 June 2014.
38. Van den Bergh, M.; Boix, X.; Roig, G.; de Capitani, B.; Van Gool, L. Seeds: Superpixels extracted via energy-driven sampling. In Proceedings of the European Conference on Computer Vision (ECCV), Florence, Italy, 7–13 October 2012.
39. Woo, K.; Kang, E.; Wang, S.; H., L.K. A new segmentation method for point cloud data. *Int. J. Mach. Tools Manuf.* **2002**, *42*, 167–178. [[CrossRef](#)]
40. Wang, M.; Tseng, Y. Automatic segmentation of LiDAR data into coplanar point clusters using an octree-based split-and-merge algorithm. *Photogramm. Eng. Remote Sens.* **2010**, *76*, 407–420. [[CrossRef](#)]
41. Yao, J.; Taddei, P.; Ruggeri, M.R.; Sequeira, V. Complex and photo-realistic scene representation based on range planar segmentation and model fusion. *Int. J. Robot. Res.* **2011**, *30*, 1263–1283. [[CrossRef](#)]
42. Rabbani, T.; Van Den Heuvel, F.; Vosselman, G. Segmentation of point clouds using smoothness constraints. In Proceedings of the ISPRS Commission V Symposium: Image Engineering and Vision Metrology, Dresden, Germany, 25–27 September 2006.
43. Cramer, M. The DGPF-test on digital airborne camera evaluation—overview and test design. *Photogramm. Fernerkund. Geoinf.* **2010**, *2010*, 73–82. [[CrossRef](#)]
44. Schnabel, R.; Wahl, R.; Klein, R. Efficient RANSAC for point-cloud shape detection. *Comput. Graph. Forum* **2007**, *26*, 214–226. [[CrossRef](#)]
45. Shan, J.; Lee, S. Quality of building extraction from IKONOS imagery. *J. Surv. Eng.* **2005**, *131*, 27–32. [[CrossRef](#)]

46. Estrada, F.; Jepson, A. Benchmarking image segmentation algorithms. *Int. J. Comput. Vis.* **2009**, *85*, 167–181. [[CrossRef](#)]
47. Kahaki, S.M.M.; Nordin, M.J.; Ashtari, A.H.; Zahra, S.J. Invariant feature matching for image registration application based on new dissimilarity of spatial features. *PLoS ONE* **2016**, *11*, e0149710.



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).