

Article

Building Extraction from Airborne LiDAR Data Based on Min-Cut and Improved Post-Processing

Ke Liu ¹ , Hongchao Ma ^{1,2,*}, Haichi Ma ¹, Zhan Cai ³ and Liang Zhang ⁴

¹ School of Remote Sensing and Information Engineering, Wuhan University, Wuhan 430079, China; Liu_ke@whu.edu.cn (K.L.); mahaichi@whu.edu.cn (H.M.)

² Department of Oceanography, Dalhousie University, Halifax, NS B3H 4R2, Canada

³ School of Resources Environment Science and Technology, Hubei University of Science and Technology, Xianning 437000, China; zhan.cai@whu.edu.cn

⁴ Faculty of Resources and Environmental Science, Hubei University, Wuhan 430062, China; zhangliang@hubei.edu.cn

* Correspondence: hchma@whu.edu.cn; Tel.: +86-1380-864-8104

Received: 18 July 2020; Accepted: 31 August 2020; Published: 2 September 2020



Abstract: Building extraction from LiDAR data has been an active research area, but it is difficult to discriminate between buildings and vegetation in complex urban scenes. A building extraction method from LiDAR data based on minimum cut (min-cut) and improved post-processing is proposed. To discriminate building points on the intersecting roof planes from vegetation, a point feature based on the variance of normal vectors estimated via low-rank subspace clustering (LRSC) technique is proposed, and non-ground points are separated into two subsets based on min-cut after filtering. Then, the results of building extraction are refined via improved post-processing using restricted region growing and the constraints of height, the maximum intersection angle and consistency. The maximum intersection angle constraint removes large non-building point clusters with narrow width, such as greenbelt along streets. Contextual information and consistency constraint are both used to eliminate inhomogeneity. Experiments of seven datasets, including five datasets provided by the International Society for Photogrammetry and Remote Sensing (ISPRS), one dataset with high-density point data and one dataset with dense buildings, verify that most buildings, even with curved roofs, are successfully extracted by the proposed method, with over 94.1% completeness and a minimum 89.8% correctness at the per-area level. In addition, the proposed point feature significantly outperforms the comparison alternative and is less sensitive to feature threshold in complex scenes. Hence, the extracted building points can be used in various applications.

Keywords: LiDAR; building extraction; normal vector estimation; min-cut; improved post-processing

1. Introduction

Automatic building extraction from remote sensing data is a prerequisite step for the applications of three-dimensional (3D) building reconstruction, urban planning, disaster assessment, and updating digital maps and geographic information system (GIS) databases [1–4]. Since the emergence of airborne Light Detection and Ranging (LiDAR), it provides an alternative way to extract buildings due to its high-density and high-accuracy point data.

In the field of LiDAR, building extraction is to separate building points from other points, which can also be termed as building detection [1,5]. Due to the complexity of building structures and urban scenes, it remains a challenge to extract buildings from LiDAR data. Many studies have been conducted in the past two decades regarding building extraction. Some methods combine other data sources, such as optical images with spectral and texture information, intensity data, waveform data

and GIS data [6–9]. Among these data, the image data is the most commonly used due to its high spatial resolution, color and texture information [6,10]. By combining two-dimensional (2D) information from images and 3D information from LiDAR data, complementary information can be exploited to extract and reconstruct buildings automatically [1,10]. However, these methods using images unavoidably involve some problems. First, LiDAR data and images need to be registered before fusion, which poses a challenge due to their different characteristics [10,11]. Second, the spatial resolution of images and LiDAR data are different, which may decrease accuracy after fusing them [10–12]. Moreover, in some regions, image and LiDAR data are not always both available to data-end users due to various reasons, which limits the practicality of these methods.

Building extraction methods based solely on point cloud are the mainstream focus [12–16]. Such methods can be broadly categorized into two classes: supervised and unsupervised methods [12,13]. Based on the basic processing units, supervised methods can be categorized into three groups: point-based, segment-based and multiple-entity-based classification methods [13,14]. Point-based classification methods are the most commonly used and are generally composed of three steps: sample points training, point features extraction, and classification. Segment-based methods generally consist of four steps: segmentation of raw points, sample segments training, segments features extraction, and classification [13]. Multiple-entity-based classification is considered as a combination of the segment-based and point-based classifications [14]. Satisfying results can be obtained from supervised approaches by using accurate features and proper training samples, but some defects cannot be ignored [15–27]. First, constant scale parameters of neighborhoods used for calculating point features fail to accurately describe local structure [21,23]. Second, most classifiers, such as JointBoost [15], support vector machine [16], random forest [17], expectation maximization [18], XGBoost [19], adaptive boosting [20], etc., classify each point independently without considering the labels of its neighborhoods, leading to inhomogeneous results in complex scenes [22,24]. Third, supervised methods involve many point features, but more features do not necessarily guarantee higher classification accuracy [27]. On the contrary, too many features may introduce redundant information and increase computation cost [25,26]. In addition, segment-based and multiple-entity-based classified methods mainly rely on the segment strategies and they are hierarchical procedures, which involve many steps [28–30].

Deep learning has been introduced in the field of point cloud in recent years, such as object recognition, classification, and segmentation [31–33]. However, the number of samples used for training models is far more than the aforementioned classifiers. Moreover, many deep learning algorithms developed for images cannot be employed for point cloud without modification due to its irregularity and discreteness [31,32].

Due to the above-mentioned reasons, many researchers study the unsupervised methods. These methods mainly include fitting methods [34–36], region growing methods [28,30], clustering methods [37,38], morphological-based methods [39,40], and energy minimization [12,15]. In the fitting methods, random sample consensus (RANSAC) [34] and 3D Hough transformation [36] methods are widely used to extract buildings. However, RANSAC approach often extracts pseudo planes in vegetation areas due to random sampling, and its efficiency significantly decreases with an increasing amount of point cloud data. 3D Hough transformation is time-consuming and sensitive to fitting parameters [36]. Region growing algorithm is also widely used due to its simple and easy operations. Generally, it only works well when the initial ideal building seeds are correctly selected and the constraints used in algorithms are reasonable. It is easy to overgrow due to unreasonable constraints and inaccurate features, especially in the transition region between different objects [28,30]. Clustering methods are statistical techniques that cluster points based on some local surface properties or other features [37,38]. In [37], building seeds were detected through semi-suppressed fuzzy C-means, then a restricted region growing algorithm was applied to search more building points based on building seeds. Morphological-based methods firstly perform rasterization on the point cloud, and then non-building pixels are removed under the constraints of size, shape, height, and building element

structure. However, the size of the pixels and the geometrical properties greatly influence the final results [39]. The energy minimization approach, such as graph cuts algorithm, is a global solution that formulates the building extraction as an optimization problem [12,15]. In [12], a graph was constructed using pixels of a generated Digital Surface Model (DSM) image, and features of points and grids were both used to construct energy function. Finally, pixels were labelled by minimizing the energy function. In [41], the graph was constructed using voxels after voxelization on point cloud. Although satisfying results can be obtained from [12,41], data structure is changed and results may exhibit decreased accuracy in the process of rasterization or voxelization due to the fact that not all points within the same pixel or voxel belong to a building, and post-processing steps are needed to solve the problem of building boundary zigzags [12]. In [15], graph cuts algorithm was used to optimize classification results of JointBoost classifier due to the fact that contextual information between points was not considered in the classifier. However, results of JointBoost classifier were used as initial foreground and background seed points to define energy function. This is impractical for automatic building extraction due to the fact that no building seeds are available before building extraction.

Based on the above statements, the current studies on building extraction still face two problems [24,27]: (1) Large numbers of point features are used to extract buildings in most existing algorithms [13,24,37]. Too many features increase computation cost of calculating point features and reduce algorithmic efficiency. Moreover, not all features are suitable for building extraction and may decrease accuracy [27]. (2) Due to various roof types and complex scenes, the contextual information is useful for classification [12,24]. However, contextual information is always ignored for most methods, leading to inhomogeneous results and increased post-processing workload.

Therefore, it is of great importance to continue building extraction research using airborne LiDAR data. The main objectives of the study are (1) to design effective point feature to discriminate building points on the intersecting roof planes from vegetation, and evaluate its performance with the existing point feature in different scenes and parameter setting; (2) to use few point features to realize building extraction, avoiding spending too much time on point features calculation; (3) to introduce contextual information in the algorithm to reduce inhomogeneity and improve accuracy of extraction results.

The main contributions of this work can be summarized as follows: (1) A point feature based on the variance of accurate normal vectors estimated via low-rank subspace clustering (LRSC) technique is proposed to discriminate building points on the intersecting roof planes from vegetation. The proposed feature significantly outperforms the comparison alternative and is less sensitive to feature threshold in complex scenes. (2) The proposed maximum intersection angle constraint effectively removes large non-building point clusters with narrow width, such as greenbelt points along streets, overcoming the defects of area-based methods in setting area threshold. (3) Contextual information and consistency constraint are both used to eliminate inhomogeneity in the proposed method, which benefits building extraction. (4) Unlike most previous building extraction methods, only two point features are used in the proposed method, which beneficially decreases the computation cost of calculating point features and improves algorithmic efficiency.

2. Methodology

The proposed method includes three main steps: outliers removal and filtering; point features calculation, graph construction and cut; and improved post-processing. The algorithm workflow is shown in Figure 1.

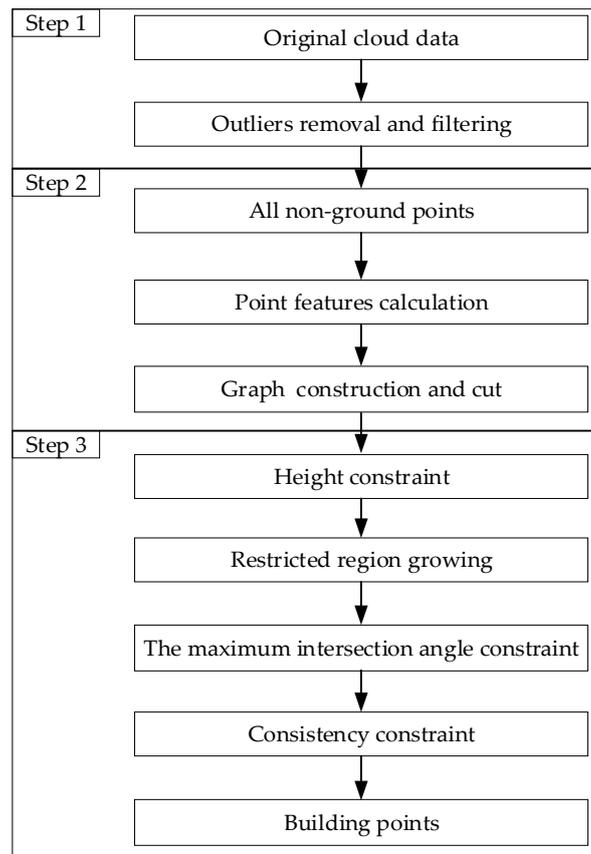


Figure 1. Algorithm flowchart.

2.1. Outliers Removal and Filtering

The original point cloud data provided by airborne LiDAR system contains some outliers obtained during data collection due to various reasons, such as multipath effect. It is necessary to remove outliers to alleviate their effects on LiDAR data processing. The “StatisticalOutlierRemoval” tool implemented in point cloud library (PCL) is applied to the original point cloud [42]. In the algorithm, the mean distance from each point to its all neighborhoods is calculated, and points whose mean distance are outside a defined range calculated by the global distance mean and standard deviation are removed [43]. After that, denoising points are classified into ground and non-ground points subsets by progressive TIN densification (PTD) [44]. PTD has been widely used in the field of academic community and in engineering applications due to its accuracy and efficiency, and it has been embedded in commercial software, such as TerraSolid and LiDAR_Suite [44,45].

2.2. Point Features Calculation and Normalization

After filtering, points can be classified as ground and non-ground points. Non-ground points are generated from laser echoes from buildings, vegetation and other man-made or natural ground objects (e.g., vehicles, wires, etc.) [15,37,40], which are the input for further building extraction process. In general, buildings are considered to be composed of planar patches, while vegetation are non-planar, meaning building points have flat characteristics and vegetation points are rough in local area. Therefore, two features based on the above characteristics are used in the proposed method.

2.2.1. Curvature Feature (f_c)

Let $NP = \{q_0, q_1, \dots, q_n\}$ denote non-ground points, and $N_p = \{p_j | p_j \in NP, p_j \in k_nearest_q_i\}$ represent the point set of k -neighborhoods of q_i . The covariance matrix M is constructed using q_i and its neighborhoods N_p , defined as follows:

$$M = \frac{1}{k} \sum_{p \in N_p} (p - \bar{p})(p - \bar{p})^T \quad (1)$$

where $\bar{p} = \frac{1}{k} \sum_{p \in N_p} p$ is the centroid of all points in N_p , and k is the number of N_p . After that, three eigenvalues $\lambda_1, \lambda_2, \lambda_3$ ($\lambda_1 \leq \lambda_2 \leq \lambda_3$) of M are calculated via eigen decomposition. The curvature feature f_c is calculated based on three eigenvalues as follows [27]:

$$f_c = \frac{\lambda_1}{\lambda_1 + \lambda_2 + \lambda_3} \quad (2)$$

The curvature feature can describe the flatness of surface, and is widely used for planes extraction [27,37,46]. Although the number of neighborhoods influences f_c , the influence is minimal [47]. Empirically k is set to 15 to calculate f_c by referring to [12].

2.2.2. Variance of LRSC-Based Normal Vector Feature (f_v)

Curvature of points on intersecting building roof planes is much larger than those of flat roofs, indicating these building points are likely to belong to vegetation [12,37]. To discriminate buildings from vegetation, feature based on the variance of normal vectors calculated via principle component analysis (PCA) is proposed in [12]. However, the estimated normal vectors via PCA without modification are inaccurate for points on intersecting building roof planes due to the fact that neighborhoods come from different planes [48]. Figure 2a,b illustrate PCA-based normal vectors of a synthetic cube and building roofs [49], and it can be seen that normal vectors of points on intersecting planes are inaccurate. As a result, features based on these normal vectors, such as variance of normal vector direction [12], and normal vector angle distribution histogram [50], are inaccurate.

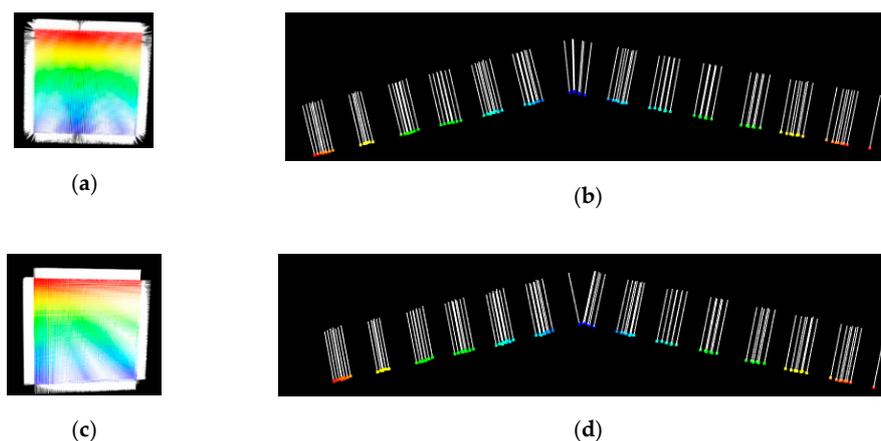


Figure 2. Normal vector. (a) PCA-based normal vector of a cube; (b) PCA-based normal vector of a building roof; (c) LRSC -based normal vector of a cube; (d) LRSC -based normal vector of a building roof.

Considering the above problem, an accurate normal vector estimation method via low-rank subspace clustering (LRSC) [48] is introduced. The normal vector estimation technique has already been used for automatic building roof segmentation from LiDAR data, and experimental results are satisfying [51]. The algorithm is composed of three main steps: First, points around sharp and smooth

regions are identified by covariance analysis of their neighborhoods, and their initial normal vectors are estimated via PCA. Second, normal vectors of points' neighborhoods are used as prior knowledge to construct a guiding matrix. Third, neighborhoods are segmented into several isotropic neighborhoods by low-rank subspace clustering (LRSC) with the guiding matrix. Then a consistent sub-neighborhood is used to estimate points' final normal vectors. Figure 2c,d illustrate the LRSC-based normal vectors of a cube and building roof. Compared with the PCA-based normal vectors, LRSC-based normal vectors of points in sharp regions are more accurate and reasonable.

Therefore, feature f_v , based on variance of LRSC-based normal vector, is proposed to discriminate buildings from vegetation, and its calculation includes following sub-steps:

Step 1: Normal vectors of all points are calculated via low-rank subspace clustering (LRSC) technique, and calculate the angle α between the normal vector and vertical direction ($\vec{v} = (0, 0, 1)$ in 3D space). It should be noted that the normal vector of points may have opposite direction. Therefore, when α is larger than $\pi/2$, the corresponding angle α is set as $\pi - \alpha$.

Step 2: For point P and angles $\alpha_P = \{\alpha_1, \alpha_2, \dots, \alpha_m\}$ of its neighborhoods $N_P = \{q_1, q_2, \dots, q_m\}$, divide the range $[0, \pi/2]$ into equal D_n bins to construct a D_n dimensional histogram. Then, the number of angles falling within each bin is taken as the value of the bin in the histogram.

Step 3: Calculate variance of histogram as f_v for P with the following formulas:

$$f_v = \frac{\sigma^2}{\mu^2} \quad (3)$$

$$\sigma^2 = \frac{\sum_{i=1}^{D_n} (n_i - \mu)^2}{D_n} \quad (4)$$

$$\mu = \frac{m}{D_n} \quad (5)$$

where m is the number of neighborhoods, n_i is the number of angles falling within the i -th bin.

Step 4: Repeat Step 2 and 3 until all points' f_v are calculated.

According to [12], the parameter m is empirically set to 60 for each point. D_n has little impact in the range of 5 to 10, and is set to 6 in the proposed method.

Figure 3a,b illustrate the feature f_v of [12] and ours respectively in Area 1 of Vaihingen [49], and points are rendered by the value of normalized f_v . Compared with [12], the proposed f_v of points on intersecting building roof planes with large slopes are almost consistent with the ones on smooth roof planes (green rectangles in Figure 3c), and difference of the proposed f_v between buildings and vegetation are more obvious (yellow rectangles in Figure 3c). In addition, it is a challenging task to discriminate some small complex buildings composed of multiple small planar patches from vegetation for f_v of [12] (blue rectangles in Figure 3c).

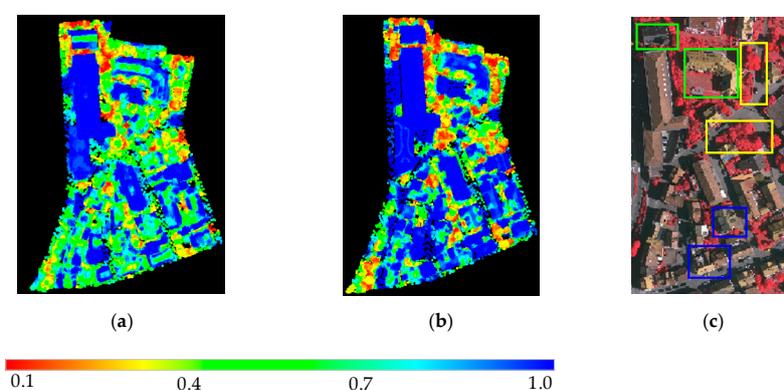


Figure 3. Visualization of f_v (rendered by the value of normalized f_v). (a) f_v of [12]; (b) the proposed f_v ; (c) Orthophoto.

2.2.3. Feature Normalization

Considering that the range of values of the two features are significantly different, a normalization step is needed. A logistic function is employed to normalize these two features, and the logistic function is defined below:

$$f(x) = \frac{1}{1 + e^{-k(x-x_0)}} \quad (6)$$

where x_0 is the feature threshold and k controls the steepness of the logistic function curve. In practice, building extraction results are significantly influenced by x_0 , and minimally influenced by k [12]. Therefore, k is set to $-35, 2.0$ for f_c and f_v respectively according to [12]. Whereas, the specific values of x_0 for f_c and f_v will be analyzed and discussed in later sections.

2.2.4. Graph Construction and Cut

Point segmentation can be viewed as a labeling problem, which is to assign a label from a set of labels to each point by minimizing an objective function [52,53]. For building extraction, the label problem is to assign a building or non-building label to each non-ground point. Generally, a typical representation of the objective function is an energy function with two terms: data term and smooth term. Among a series of optimization methods to minimize energy function, graph cuts approach [54] based on minimum cut (min-cut) shows good performance since it merges, and is commonly used in many applications, such as image segmentation [55–57] and point cloud segmentation [12,15,41]. Thus, graph cuts algorithm is adopted to minimize energy function.

In [54], the graph is composed of sets of nodes and edges. It should be noted that there are two special terminal nodes, called source and sink, which represent the “foreground” and “background” labels. In the proposed method, each non-ground point denotes a node. The energy function $E(l)$ is defined as follows:

$$E(l) = \sum_{p \in P} D_p(l_p) + \sum_{p, q \in N} V_{pq}(l_p, l_q) \quad (7)$$

where the first term $\sum_{p \in P} D_p(l_p)$ is a data term and $D_p(l_p)$ is the penalty to assign label l_p to node p . Value of $D_p(l_p)$ measures how well the label fits node p . The second term $\sum_{p, q \in N} V_{pq}(l_p, l_q)$ is the smooth term and $V_{pq}(l_p, l_q)$ is interpreted as a penalty for discontinuity between nodes p and q . Generally, if p and q are similar, $V_{pq}(l_p, l_q)$ is large, which means p and q more likely belong to the same object. Data penalty $D_p(l_p)$ is calculated as follows:

$$D_p(l_p = \text{building}) = \lambda_1 f_c + \lambda_2 f_v \quad (8)$$

$$D_p(l_p = \text{non - building}) = 1 - (\lambda_1 f_c + \lambda_2 f_v) \quad (9)$$

where λ_1 and λ_2 are the weights of f_c and f_v respectively, and they satisfy $\lambda_1 + \lambda_2 = 1$.

Smooth penalty $V_{pq}(l_p, l_q)$ is calculated as follows:

$$V_{pq}(l_p, l_q) = e^{-(\lambda_1 |f_{pc} - f_{qc}| + \lambda_2 |f_{pv} - f_{qv}|)} \cdot \frac{1}{d(p, q)^2} \quad d(p, q) < d_s \quad (10)$$

$$V_{pq}(l_p, l_q) = e^{-(\lambda_1 |f_{pc} - f_{qc}| + \lambda_2 |f_{pv} - f_{qv}|)} \cdot \frac{1}{d_s^2} \quad d(p, q) \geq d_s \quad (11)$$

where f_{pc} and f_{qc} are f_c of p and q , f_{pv} and f_{qv} are f_v of p and q . $d(p, q)$ is the Euclidean distance between point p and q . $|\cdot|$ is any norm distance metric and L1 norm is adopted, d_s is the distance threshold between points and is set as twice the average point space. When the graph is constructed, it is cut based on min-cut and each node will be given a label. Thus, building points are extracted according to the given labels.

2.3. Improved Post-Processing

Although most building points are successfully extracted from the non-ground points, some non-building points are wrongly classified as buildings (e.g., vehicles with smooth surfaces and flat overpasses) and some building points are omitted. To solve these aforementioned problems, an improved post-processing is adopted to refine results of building extraction.

2.3.1. Height Constraint

In general, a building should be high enough. Thus, height threshold T_h is set to remove these low points if their absolute height difference between it and its nearest ground points is less than T_h . Under this constraint, partial or possibly whole points of vehicles with smooth surfaces can be excluded due to their low height. T_h is set according to the average human's height, such as 1.5 m [23].

2.3.2. Restricted Region Growing

It should be noted that some buildings are located on a slope, and some points satisfy the height constraint. Figure 4a illustrates a profile of a building located in a slope, and partial points are classified as non-building points after height constraint. Thus, restricted region growing based on height constraint is conducted to extract omitted buildings. In the process of restricted region growing, the non-building points are classified as buildings if the absolute height difference between them and their nearest building points is less than a predefined threshold T'_h . T'_h is set to 0.1 m according to [37], and building extraction results after restricted region growing are shown in Figure 4b. It should be noted that ground, building and non-building points are rendered by blue, red, and white respectively in subsequent sections.

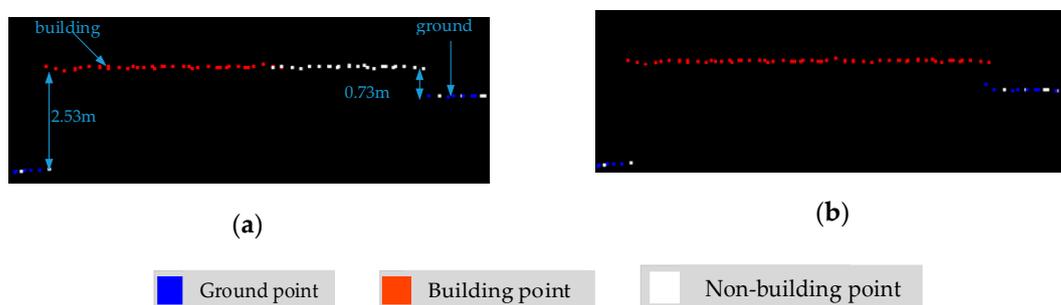


Figure 4. Profile of a building located on a slope. (a) Before restricted region growing; (b) After restricted region growing.

2.3.3. Maximum Intersection Angle Constraint

After the above-mentioned steps, some non-building points or clusters belonging to vegetation with flat surface, or vehicles with smooth surface and small size are wrongly classified as buildings. Area-based strategies are commonly adopted to remove these above non-building points by clustering based on the assumption that buildings generally occupy a specific area [12,23,27]. Although satisfying results can be obtained by setting proper area thresholds, they fail to eliminate non-building points in some scenes. Figure 5 illustrates building extraction results in Area 3 of Vaihingen [49] after the above steps from LiDAR data with an average point density of 3.2 points/m². Building points are separated into two clusters via Euclidean clustering, in which one is located in green and the other in the yellow rectangle. The number of points of the small cluster in the yellow rectangle is more than 110, which means the cluster occupies approximate 35 m². But in practice, the area of many buildings is less than that and they will be eliminated via area-based methods if the area threshold is set to 35 m². To solve this issue, the maximum intersection angle constraint is proposed.

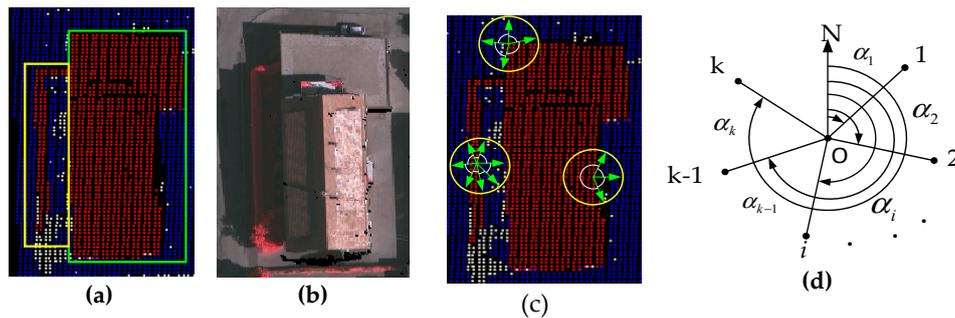


Figure 5. Concept of intersection angle. (a) Building extraction after restricted region growing; (b) Orthophoto; (c) Concept of intersection angles; (d) Calculation of the maximum intersection angle.

Figure 5c illustrates the concept of intersection angle, which is composed of the current building point and its cylindrical neighborhoods searched from non-building and ground points. Due to the fact that buildings occupy a specific area and the façade of buildings act as barriers to prevent ground points falling inside the building area, the maximum intersection angle of real building points is larger than 90° at the building corners, and larger than 180° away from building corner. While, the maximum intersection angle of vegetation points or other non-building points is less than 90° due to the fact that it is surrounded by ground and non-building points in all directions within a cylinder (yellow circle in Figure 5c) [58].

Figure 5d shows the calculation of the maximum intersection angle of a given current building point O and its cylindrical nearest neighborhoods. The calculation is composed of three main steps:

Step 1: select initial direction \vec{ON} and calculate rotational angle α_i with respect to \vec{ON} using their x, y coordinates for each nearest point.

Step 2: sort above angles in ascending order, and intersection angle δ between adjacent rotational angles is calculated as follows:

$$\delta_i = \begin{cases} \alpha_{i+1} - \alpha_i & i = 1, 2, \dots, k-1 \\ 360 + \alpha_1 - \alpha_k & i = k \end{cases} \quad (12)$$

Step 3: the maximum δ is selected as the maximum intersection angle of point O.

For a current building point O, if its maximum intersection angle is larger than the pre-defined threshold, then O is classified as a building point. Otherwise, it is a non-building point. If there are neither non-building points nor ground points falling in the cylinder of O, then it is classified as buildings directly.

The maximum intersection angle constraint takes into account two threshold parameters: radius of cylinder to detect neighborhoods from ground and non-building points and angular threshold to consider the minimum angle defined by the façade alignments at the corners. In the proposed method, the radius is empirically set to 2.5 m according to [12,37] and the angular threshold was set to 90° according to [58].

It should be noted that generally the width of a flat overpass ranges between 2.5 m and 3.5 m [59], thus flat overpasses can be excluded from the detected building points under the constraint of the maximum intersection angle using above empirical thresholds.

2.3.4. Consistency Constraint

Although most buildings are extracted after the above three steps, some special building points are omitted and some non-building points are wrongly classified as buildings. Figure 6a illustrates a roof terrace with some attachments in Vaihingen [49], such as tables, chairs, and small vegetation on it. It is obvious that points in this area are rough, while the building surface is flat, as shown in Figure 6c. As a result, some building points fail to be detected (hereinafter referred to as undetected building

points). Figure 7 shows two trees with dense leaves and the top of one is flat. Consequently, points falling in the region are wrongly classified as buildings (hereinafter referred to as false building points).

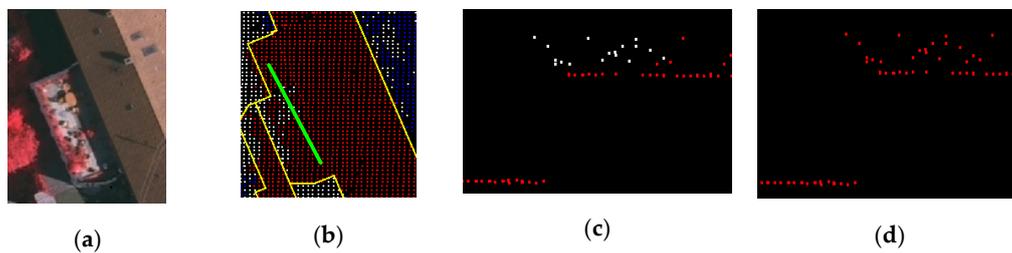


Figure 6. Roof terrace. (a) Orthophoto; (b) Point data. Yellow lines are reference vectors provided by ISPRS. The approximate location of profile is indicated by the green line; (c) Profile before consistency constraint; (d) Profile after consistency constraint.

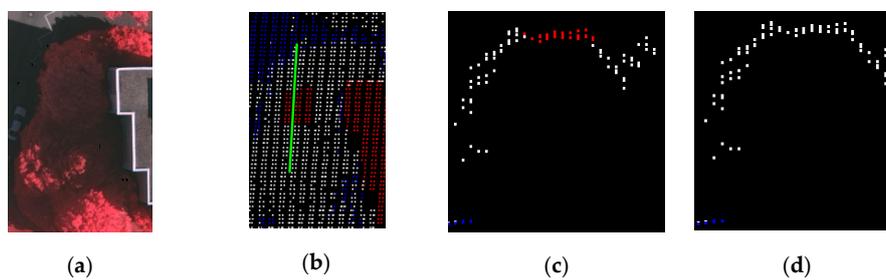


Figure 7. Two trees with dense leaves. (a) Orthophoto; (b) Point data. The approximate location of profile is indicated by the green line; (c) Profile before consistency constraint; (d) Profile after consistency constraint.

It should be noted that these undetected building points (false building points) are surrounded by building points (non-building points). Therefore, consistency constraint is proposed to solve the problem.

Considering that the steps of detecting undetected building points and removing false building points are similar, we take the elimination of false building points as example to introduce the process. It is composed of three main steps: First, the minimal and maximal values of x and y coordinates of non-ground points are obtained and denoted by $x_{max}, x_{min}, y_{max}, y_{min}$. Then the minimum bounding rectangle that covers non-building and ground points is partitioned into uniform cells with size l . Second, for each building point $p(x_i, y_i)$, row number $[(x_i - x_{min})/l]$ and column number $[(y_i - y_{min})/l]$ falling in cell g are calculated using its x, y coordinates. Third, search cells in a direction and stop if a cell contains points. Figure 8 illustrates search path in four directions. Where cell g is rendered by red, a yellow arrow with the same direction shows the search path in this direction, the last cell in a direction is colored by magenta, other cells are render by green. The l is set as twice the average point space according to [12,37].

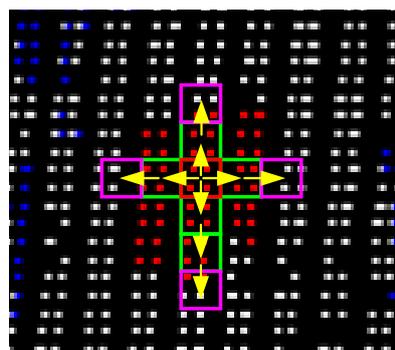


Figure 8. Search path in four directions.

Finally, 4 non-empty cells are obtained. If there are non-building points and no ground points in these 4 cells, then $p(x_i, y_i)$ belongs to non-building points. Otherwise, it belongs to buildings. Figure 6d illustrates the undetected building points are extracted and Figure 7d shows the false building points are removed, avoiding inhomogeneity in the results of building extraction.

3. Experimental Results and Analysis

To validate the proposed method, datasets provided by the International Society for Photogrammetry and Remote Sensing (ISPRS) in Vaihingen and Toronto [49], one dataset in New Zealand [60] and one dataset in the state of Utah [61], were used in the experiments. The results were displayed in LiDAR_Suite, an airborne LiDAR data processing software developed by the Research and Development (R&D) group of the authors.

3.1. Experiments on the ISPRS Benchmark Dataset

3.1.1. Data Description

The ISPRS dataset is composed of five reference areas: Area 1 to 3 in Vaihingen and Area 4 to 5 in Toronto. The average point density of Area 1 to 3 is approximately 4–7 points/m², and it is about 6 points/m² in Area 4 to 5. It should be noted that three test areas (i.e., Area 1 to 3) are located in residential regions and other two test areas (i.e., Area 4 to 5) are within commercial zones, as shown in Figure 9a–e.

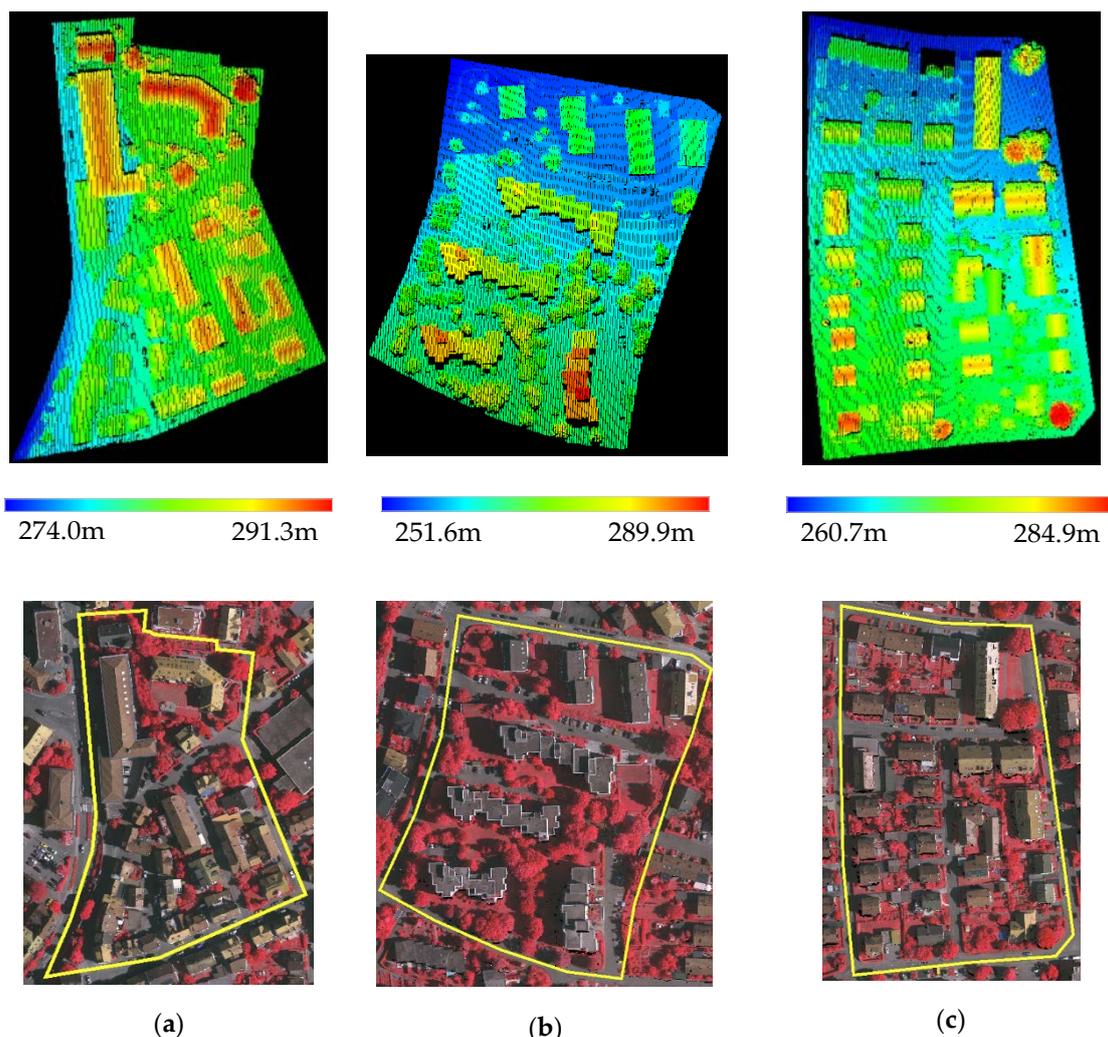


Figure 9. Cont.

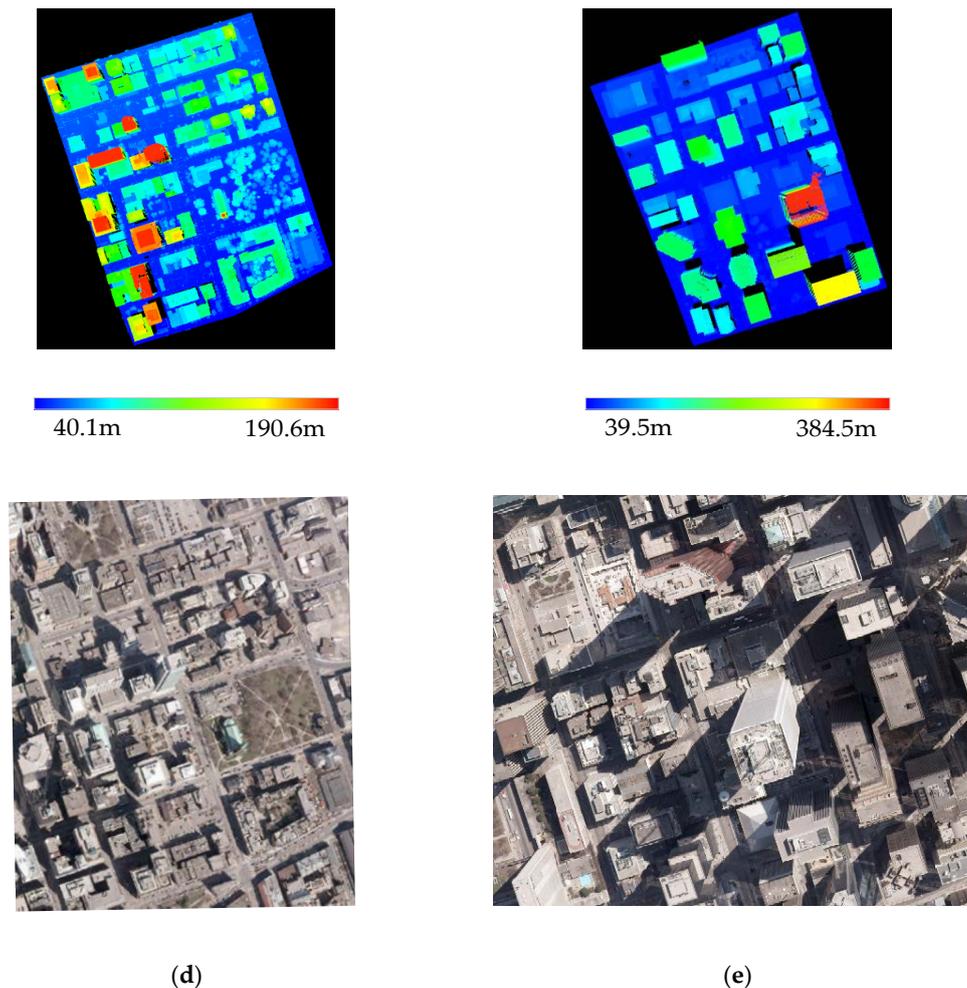


Figure 9. Test datasets. (a) Area 1; (b) Area 2; (c) Area 3; (d) Area 4; (e) Area 5.

Area 1: includes 37 buildings, mainly composed of dense historic buildings having rather complex shapes along with roads and trees.

Area 2: includes 14 buildings, mainly composed of a few high regular residence buildings with horizontal roofs.

Area 3: includes 56 buildings, mainly composed of several detached buildings with simple structured roofs and vegetation along roads.

Area 4: includes 58 buildings, mainly composed of low and high-storey buildings with complex roof structure and rooftop attachments.

Area 5: includes 38 buildings, including a cluster of high-rise buildings with diverse roof structure, complex shapes and rooftop attachments.

3.1.2. Results and Analysis

Initial results of building extraction based on graph cuts algorithm and height constraint are shown in Figure 10. It can be seen that most buildings are successfully extracted in Area 1 to 5 and almost no inhomogeneity exists for individual buildings in Area 1 to 3 where there are almost no irregular complex attachments on roofs, due to the fact the proposed method considers spatial neighborhood information, which helps to exploit contextual information and improve classification accuracy [41,62]. The satisfying initial results benefit post-processing work. However, small clusters or individual points belonging to vegetation or small objects, still exist, as shown in the green rectangles in Figure 10b,c. Also, few points inside the buildings fail to be extracted, as shown in green rectangles in Figure 10a.

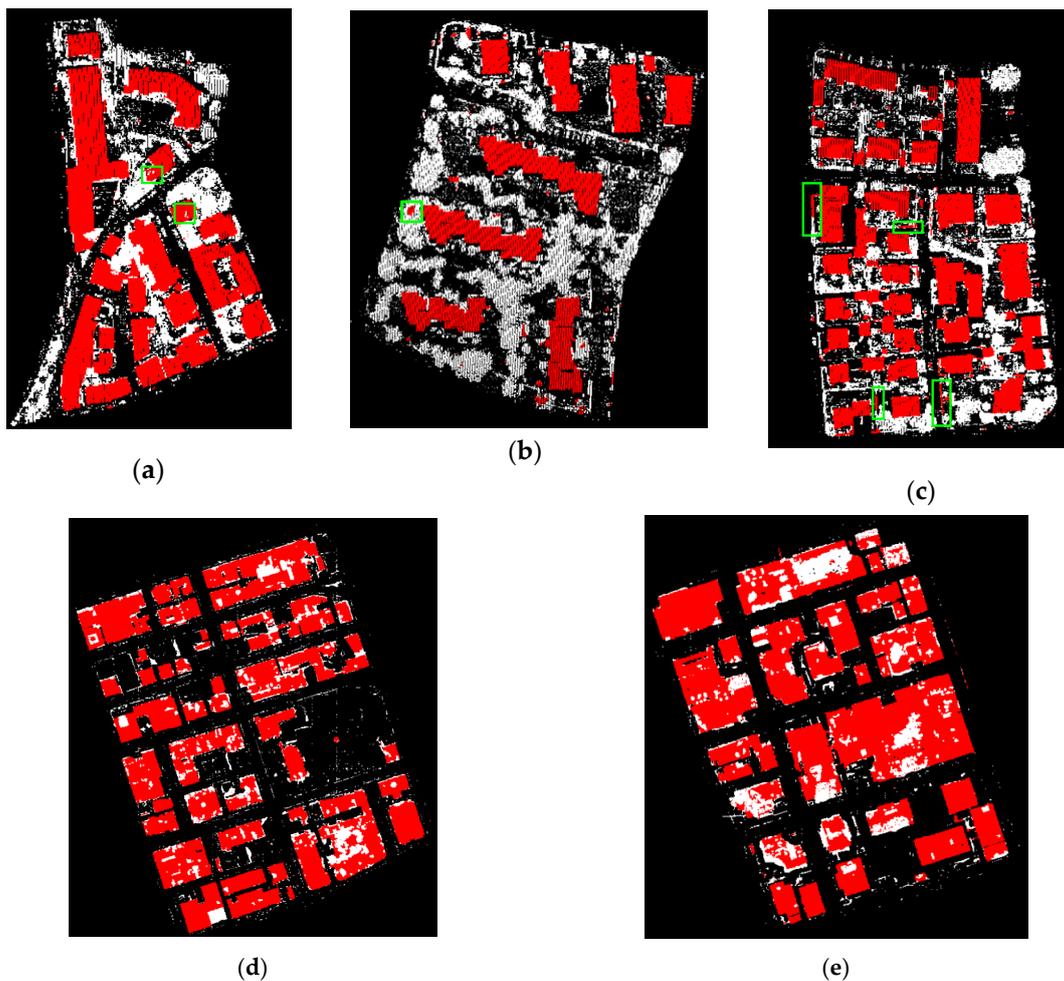


Figure 10. Initial results of building extraction based on graph cuts and height constraints. (a) Results of Area 1; (b) Results of Area 2; (c) Results of Area 3; (d) Results of Area 4; (e) Results of Area 5.

For qualitative evaluation, four indicators: completeness CP (%), correctness CR (%), quality Q (%) and F_1 -score F_1 (%) metrics of the results on per-area and per-object levels are evaluated [37,63–65]. Completeness represents the percentage of correctly detected buildings to total number of reference buildings. Correctness denotes the percentage of correctly detected buildings to total number of detected buildings. Quality and F_1 -score provide a compound performance metric that balances completeness and correctness. Equations are described as follows:

$$\begin{aligned}
 CP &= \frac{TP}{TP+FN} \\
 CR &= \frac{TP}{TP+FP} \\
 Q &= \frac{TP}{TP+FN+FP} \\
 F_1 &= \frac{2*CP*CR}{CP+CR}
 \end{aligned} \tag{13}$$

where TP is the number of correctly detected buildings, FN is the number of omitted buildings, and FP is the number of wrongly detected buildings. The quantity evaluation results of Area 1 to 5 are listed in Table 1. To compare the performances of different methods to extract buildings from Area 1 to 3, 20 methods [12,23,37,49] that solely use LiDAR data are selected and the average results are listed in Table 2, where the ID “WHU_TQ” refers to the proposed method. For Area 4 to 5, 11 methods [23,49] are selected and compared with the proposed method, listed in Table 3. Figure 11 demonstrates the extraction results in the pixel against the reference ground truth data.

From Tables 1–3, we find that satisfying results on a per-area and per-object level are obtained by the proposed method. The CP metric at the area level is important because it is directly related to the post-editing work and subsequent building modeling. At this level, 95.5% and 98.4% of average CP in Vaihingen and Toronto are obtained respectively, outperforming other comparison methods in Tables 2 and 3 respectively, which means buildings are more easily recognized by the proposed method. In addition, differences in Q between these five areas are minimal at the area level, which demonstrates the stability of the proposed method. Although the average Q metric of Area 1 to 3 on a per-area level is less than some methods, such as [12,23,37], there were some defects for them. Contextual information was not considered in [23,37], while existing research works have demonstrated that contextual information is beneficial to data processing [12,15,24]. As a result, the initial extraction results of [37] are not as good as ours due to lack of contextual information, which increase post-processing work. In [23], some inhomogeneity occurred in the process of extracting complex buildings. Moreover, the number of point features used in the proposed method is less than [12,23,37], which benefits building extraction by decreasing computation cost of calculating point features and improving algorithmic efficiency. For Area 4 to 5 datasets, the average Q metrics of the best method slightly outperforms the proposed method at the area level. This is mainly because there are several non-building objects with flat surfaces and large size, and they are wrongly classified as buildings. But the CP of the proposed method is obviously higher than other methods at the per-area level.

From Tables 2 and 3, it can be seen that scores of four metrics are different for the same method, indicating different methods should be chosen merely according to different metrics. In fact, the best method should be application-oriented. For example, building with floor area larger than 50 m² are more important in urban planning and reconstruction [1,5], thus it is reasonable to choose [12] to extract buildings from Area 1 to 3 datasets. Efficiency of a method needs to be considered in practice as well.

Table 1. Building extraction results. (The best values per column are shown in the bold font.).

| Test Case | Per-Area (%) | | | | Per-Object (%) | | | | Per-Object > 50 m ² (%) | | | |
|-----------|--------------|-------------|-------------|----------------|----------------|------------|-------------|----------------|------------------------------------|------------|------------|----------------|
| | CP | CR | Q | F ₁ | CP | CR | Q | F ₁ | CP | CR | Q | F ₁ |
| Area 1 | 97.1 | 91.4 | 88.9 | 94.2 | 83.8 | 96.9 | 81.6 | 89.9 | 100 | 100 | 100 | 100 |
| Area 2 | 95.4 | 92.9 | 88.9 | 94.1 | 85.7 | 100 | 85.7 | 92.3 | 100 | 100 | 100 | 100 |
| Area 3 | 94.1 | 90.2 | 85.4 | 92.1 | 83.9 | 100 | 83.9 | 91.2 | 97.4 | 100 | 97.4 | 98.7 |
| Average | 95.5 | 91.5 | 87.7 | 93.5 | 84.5 | 99.0 | 83.7 | 91.2 | 99.1 | 100 | 99.1 | 99.5 |
| Area 4 | 98.2 | 90.5 | 89.1 | 94.2 | 98.3 | 85.5 | 84.6 | 91.5 | 100 | 93.4 | 93.4 | 96.6 |
| Area 5 | 98.6 | 89.8 | 88.7 | 94.0 | 94.7 | 72.0 | 69.2 | 81.8 | 97.1 | 84.6 | 82.5 | 90.4 |
| Average | 98.4 | 90.2 | 88.9 | 94.1 | 96.5 | 78.8 | 76.7 | 86.6 | 98.6 | 89.0 | 88.0 | 93.5 |

Table 2. Average evaluation results comparison of the building extraction (Area 1 to 3). (The best values per column are shown in the bold font.).

| ID | Per-Area (%) | | | | Per-Object (%) | | | | Per-Object > 50 m ² (%) | | | |
|-------|--------------|------|------|----------------|----------------|-------|------|----------------|------------------------------------|-------|-------|----------------|
| | CP | CR | Q | F ₁ | CP | CR | Q | F ₁ | CP | CR | Q | F ₁ |
| UMTA | 92.3 | 87.5 | 81.5 | 89.8 | 80.0 | 98.6 | 79.1 | 88.3 | 99.1 | 100.0 | 99.1 | 99.5 |
| UMTP | 92.4 | 86.0 | 80.3 | 89.1 | 80.9 | 95.8 | 78.1 | 87.7 | 98.8 | 97.2 | 96.0 | 98.0 |
| MON | 92.7 | 88.7 | 82.8 | 90.7 | 82.7 | 93.1 | 77.7 | 87.6 | 99.1 | 100.0 | 99.1 | 99.5 |
| VSK | 85.8 | 98.4 | 84.6 | 91.7 | 79.7 | 100.0 | 79.7 | 88.7 | 97.9 | 100.0 | 97.9 | 98.9 |
| WHUY1 | 87.3 | 91.6 | 80.8 | 89.4 | 77.6 | 98.1 | 76.5 | 86.7 | 97.4 | 97.9 | 95.4 | 97.6 |
| WHUY2 | 89.7 | 90.9 | 82.3 | 90.3 | 83.0 | 97.5 | 81.3 | 89.7 | 99.1 | 98.0 | 97.2 | 98.5 |
| HANC1 | 91.5 | 92.5 | 85.2 | 92.0 | 81.5 | 72.7 | 62.4 | 76.8 | 100.0 | 95.8 | 95.8 | 97.9 |
| HANC2 | 90.2 | 93.2 | 84.6 | 91.7 | 85.1 | 69.6 | 61.9 | 76.6 | 100.0 | 100.0 | 100.0 | 100.0 |
| MAR1 | 87.0 | 97.1 | 84.8 | 91.8 | 78.2 | 96.2 | 75.7 | 86.3 | 99.1 | 100.0 | 99.1 | 99.5 |
| MAR2 | 89.7 | 95.2 | 85.8 | 92.4 | 80.6 | 93.7 | 76.5 | 86.7 | 99.1 | 98.9 | 98.0 | 99.0 |
| TON | 77.7 | 97.7 | 76.3 | 86.6 | 67.5 | 98.9 | 66.9 | 80.2 | 92.7 | 98.8 | 91.6 | 95.7 |
| HANC3 | 91.3 | 95.9 | 87.8 | 93.5 | 85.4 | 82.2 | 71.7 | 83.8 | 100.0 | 98.9 | 98.9 | 99.4 |

Table 2. Cont.

| ID | Per-Area (%) | | | | Per-Object (%) | | | | Per-Object > 50 m ² (%) | | | |
|--------|--------------|-------------|-------------|----------------|----------------|--------------|-------------|----------------|------------------------------------|--------------|--------------|----------------|
| | CP | CR | Q | F ₁ | CP | CR | Q | F ₁ | CP | CR | Q | F ₁ |
| WHU_QC | 85.8 | 98.7 | 84.8 | 91.8 | 80.9 | 99.0 | 80.3 | 89.0 | 96.8 | 100.0 | 96.8 | 98.4 |
| MON2 | 87.6 | 91 | 80.6 | 89.3 | 86.3 | 93.9 | 81.6 | 89.9 | 99.1 | 100.0 | 99.1 | 99.5 |
| WHU_YD | 89.8 | 98.6 | 88.6 | 94.0 | 87.8 | 99.3 | 87.3 | 93.2 | 99.1 | 100.0 | 99.1 | 99.5 |
| MON4 | 94.3 | 82.9 | 79.0 | 88.2 | 83.9 | 93.8 | 79.3 | 88.6 | 99.1 | 100.0 | 99.1 | 99.5 |
| MON5 | 89.9 | 90.3 | 82 | 90.1 | 87.2 | 96.3 | 84.4 | 91.5 | 99.1 | 100.0 | 99.1 | 99.5 |
| [12] | 94.0 | 94.9 | 89.5 | 94.4 | 83.3 | 100.0 | 83.3 | 90.9 | 100.0 | 100.0 | 100.0 | 100.0 |
| [23] | 89.8 | 98.6 | 88.6 | 94.0 | 87.8 | 99.3 | 87.3 | 93.2 | - | - | - | - |
| [37] | 93.4 | 95.8 | 89.6 | 94.6 | - | - | - | - | - | - | - | - |
| WHU_TQ | 95.5 | 91.5 | 87.7 | 93.5 | 84.5 | 99.0 | 83.7 | 91.2 | 99.1 | 100.0 | 99.1 | 99.5 |

Table 3. Average evaluation results comparison of the building extraction (Area 4 to 5). (The best values per column are shown in the bold font.)

| ID | Per-Area (%) | | | | Per-Object (%) | | | | Per-Object > 50 m ² (%) | | | |
|--------|--------------|-------------|-------------|----------------|----------------|-------------|-------------|----------------|------------------------------------|-------------|-------------|----------------|
| | CP | CR | Q | F ₁ | CP | CR | Q | F ₁ | CP | CR | Q | F ₁ |
| TUM | 85.1 | 80.6 | 70.6 | 82.7 | 83.9 | 90.3 | 76.9 | 87.0 | 88.2 | 92.5 | 82.3 | 90.3 |
| MAR1 | 96.1 | 92.1 | 88.7 | 94.0 | 98.7 | 86.8 | 85.8 | 92.4 | 98.6 | 87.6 | 86.5 | 92.8 |
| WHUY2 | 94.3 | 91.3 | 86.5 | 92.7 | 90.4 | 95.8 | 87.0 | 93.0 | 94.8 | 95.8 | 91.0 | 95.3 |
| ITCM | 76.9 | 87.5 | 69.2 | 81.8 | 86.5 | 21.7 | 20.9 | 34.6 | 89.7 | 70.5 | 65.2 | 78.9 |
| ITCR | 75.0 | 94.5 | 71.9 | 83.6 | 79.6 | 43.5 | 39.1 | 56.2 | 83.8 | 91.8 | 77.9 | 87.6 |
| MAR2 | 94.0 | 94.3 | 88.9 | 94.1 | 91.3 | 91.9 | 84.5 | 91.6 | 95.7 | 96.8 | 92.8 | 96.2 |
| MON2 | 95.9 | 92.2 | 88.7 | 94.0 | 93.4 | 81.1 | 76.7 | 86.8 | 95.7 | 94.5 | 90.7 | 95.1 |
| Z_GIS | 91.7 | 90.3 | 83.4 | 91.0 | 95.7 | 86.4 | 83.1 | 90.8 | 96.3 | 87.3 | 84.4 | 91.5 |
| WHU_YD | 95.8 | 94.6 | 90.8 | 95.2 | 91.3 | 95.4 | 87.4 | 93.3 | 95.7 | 95.4 | 91.4 | 95.5 |
| HKP | 97.6 | 92.7 | 90.6 | 95.1 | 93.9 | 90.4 | 85.4 | 92.1 | 95.7 | 90.4 | 86.9 | 93.0 |
| [23] | 95.8 | 94.7 | 90.8 | 95.2 | 91.3 | 95.4 | 87.5 | 93.3 | - | - | - | - |
| WHU_TQ | 98.4 | 90.2 | 88.9 | 94.1 | 96.5 | 78.8 | 76.7 | 86.6 | 98.6 | 89.0 | 88.0 | 93.5 |

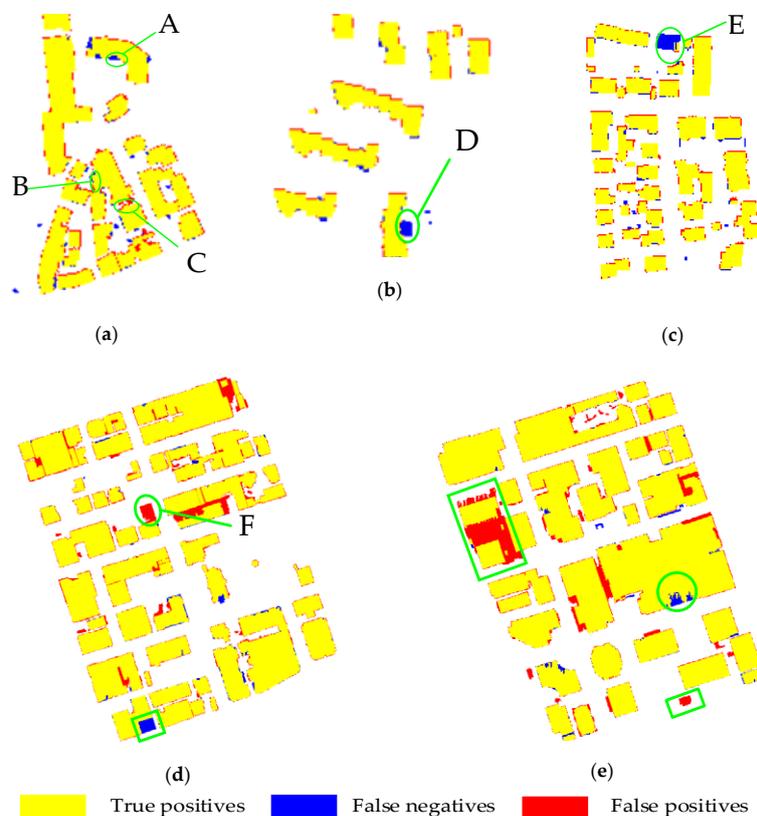


Figure 11. Cont.

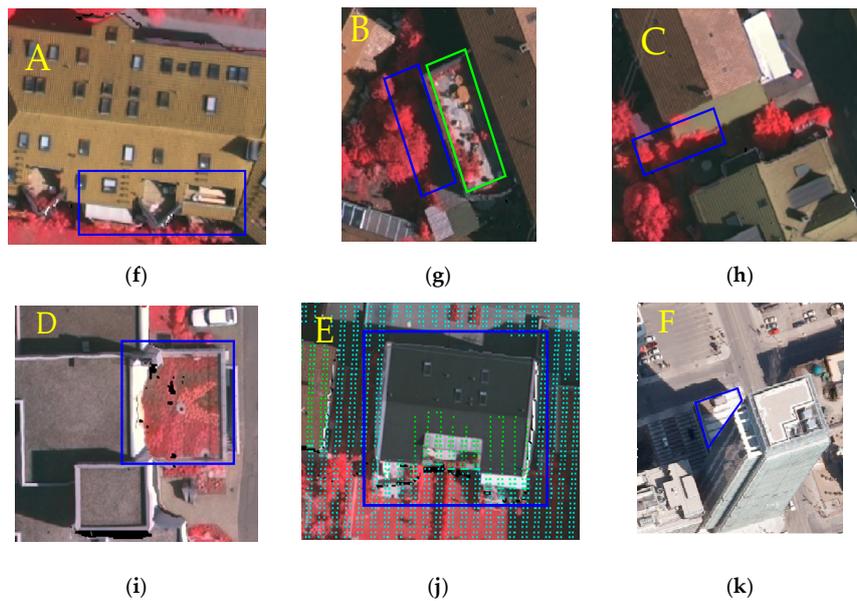


Figure 11. Visualization of building extraction results at the per-pixel level and error factors. (a) Area 1; (b) Area 2; (c) Area 3; (d) Area 4; (e) Area 5; (f) Details of A in Area 1; (g) Details of B in Area 1; (h) Details of C in Area 1; (i) Details of D in Area 2; (j) Details of E in Area 3; (k) Details of F in Area 4.

From Figure 11, most buildings can be correctly extracted by the proposed method. Although some vegetation with smooth surfaces are wrongly classified as buildings (green rectangle in Figure 10b), they are eliminated under the consistency constraint. Also, there are lots of long greenbelts vegetation area with the same height above 1.5 m, and points of these long greenbelts are classified as buildings due to their flat surfaces (shown in green rectangle in Figure 10c). These are eliminated using the maximum intersection angle constraint. However, some building attachments are omitted (hereinafter referred to as false negative errors) and some non-building points are wrongly classified as buildings (hereinafter referred to as false positive errors).

False negative errors can be explained by four main reasons: (1) Complex structure of buildings. In Figure 11f, there is a skylight (about 3.0 m * 4.0 m) and a roof terrace (about 2.5 m * 3.0 m) containing some small attachments (seems to be chairs, bed). As a result, points in these regions are rough and it is hard to effectively extracted buildings, which is a common problem for many methods [12,23,37]. However, if the attachments locate in large roof terrace, then the roof terrace can be extracted under consistency constraint, which is demonstrated in the green rectangle region in Figure 11g. The phenomenon also occurs in Area 5, shown in green circle in Figure 11e. (2) Occlusion by vegetation. The blue rectangle in Figure 11g is a building roof, and partial buildings are occluded. As a result, partially non-occluded buildings are extracted successfully. When a small building is partially occluded by vegetation, then the whole individual building is omitted. (3) Building with low height. Generally, a building should be high enough for people going in and out. However, in Figure 11i, the maximal absolute height difference between building and ground is about 1.0 m, less than 1.5 m, and many building points' height are almost equal to ground. In [12,41], points in Figure 11i were wrongly classified as ground due to the above reasons. In this case, buildings failed to be extracted. Actually, without height constraint, points in this region are extracted in our experiments. The phenomenon also occurs in Area 4, shown in green rectangle in Figure 11d. (4) Data missing. Possibly due to the building roof material shown in Figure 11j, partial point cloud data of building are collected. Thus, partial buildings are extracted by the proposed method. A possible solution to solve the problem is to fuse LiDAR data and images.

Conversely, false positive errors occur for two main reasons: (1) Surrounding vegetation with height similar to building and smooth surface. Due to the fact that graph cuts algorithm considers

the neighborhood relationship based on feature difference and the features are calculated using neighborhoods, then building regions are easy to overspread to partial vegetation areas, shown in Figure 11h. Although the problem can be solved by decreasing smooth term, completeness will decrease for lack of neighborhood information. (2) Non-building objects similar to buildings. Figure 11k illustrates a non-building object with flat surfaces and large size, and it is hard to discriminate them from buildings merely according to commonly used constraints, such as height constraint, area constraint, and the maximum intersection angle constraint. As a result, it is wrongly classified as building. This phenomenon also occurs in Area 5 in green rectangle in Figure 11e. A possible solution to this problem is the fusion of LiDAR data and other data sources.

3.2. Experiments on Other Two LiDAR Datasets

3.2.1. Data Description

Two datasets with different point density are used, one of which is captured in New Zealand [60] and the other which is captured in state of Utah [61]. The average point density of New Zealand dataset is about 20 points/m², while Utah dataset is about 3 points/m², as shown in Figure 12.

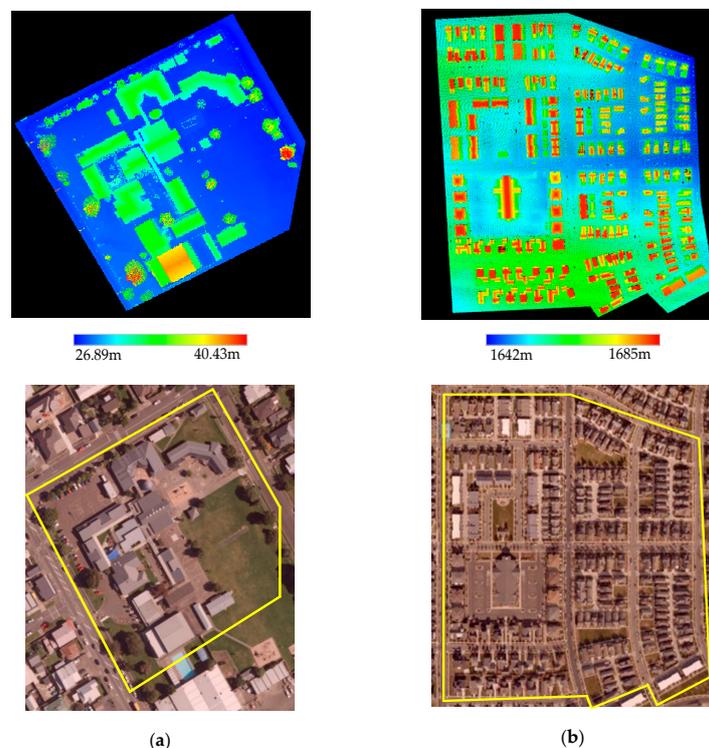


Figure 12. Test datasets. (a) New Zealand dataset; (b) Utah dataset.

New Zealand dataset: includes two buildings with curved roofs and several large connected complex buildings.

Utah dataset: includes dense residential buildings with significantly different sizes, shapes and structures surrounded by vegetation.

3.2.2. Results and Analysis

New Zealand dataset and Utah dataset have been classified into four classes: ground, vegetation, building, and others by using LiDAR_Suite and manual post-processing. Therefore, the obtained buildings are used as truth data in the experiments. Building extraction results at the per-pixel level of

these two datasets are illustrated in Figure 13, and quantitative evaluation results of building extraction at the area level are shown in Table 4.

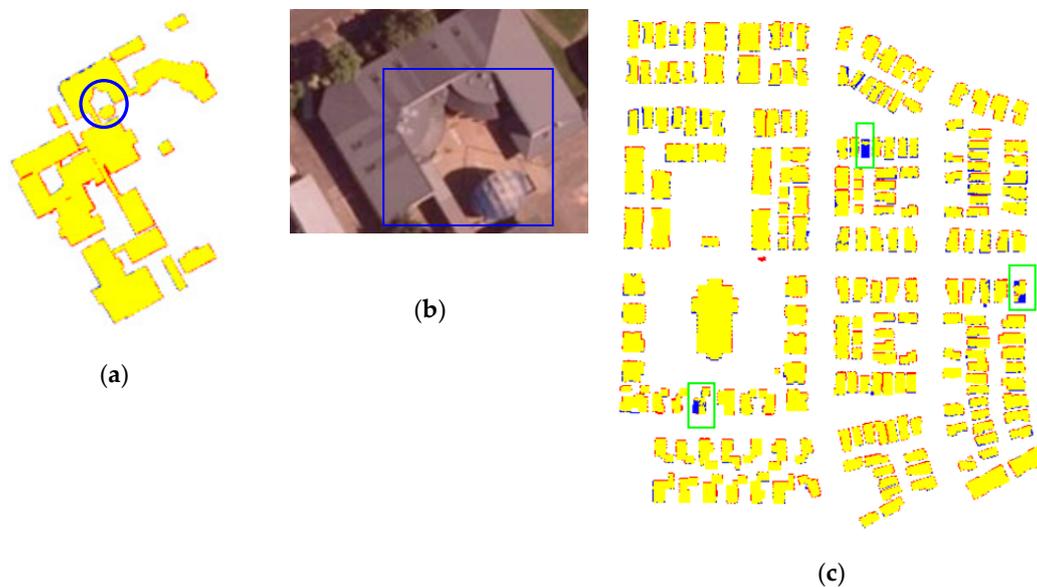


Figure 13. Building extraction at the per-pixel level. (a) Results of New Zealand dataset; (b) Details in New Zealand dataset; (c) Results of Utah dataset.

Table 4. Evaluation results of building extraction at the area level.

| Data | Precision (Per-Area) | CP (%) | CR (%) | Q (%) | F_1 |
|---------------------|----------------------|--------|--------|-------|-------|
| | | | | | |
| New Zealand dataset | | 98.4 | 94.7 | 93.2 | 96.5 |
| Utah dataset | | 95.3 | 92.3 | 88.3 | 93.8 |

In New Zealand dataset, the large complex connected buildings and the buildings with curved roofs are successfully extracted, as shown in Figure 13a,b. It should be noted that the Q (%) in New Zealand dataset is 93.2%, significantly larger than Utah dataset with 88.3%. The reason may include: (1) New Zealand dataset is high-density point data with 20 points/m², and more accurate details can be obtained when point density increases [47], which is beneficial to building extraction; (2) Buildings are far from vegetation and slightly occluded by vegetation.

In Utah dataset, although the sizes, shapes and structures of buildings are significantly different, most buildings are successfully extracted by the proposed method. However, three buildings (green rectangles in Figure 13c) fail to be extracted completely, due to two main possible reasons: (1) Data missing. Possibly due to the building roof material, partial building points of these three buildings are collected, as shown in Figure 12b. As a result, points are rough in these local areas and partial points are classified as buildings and others are not. (2) The maximum intersection angle constraint. Some non-building and building points mix together after performing graph cuts algorithm due to missing data, then these building points are eliminated under the maximum intersection angle constraint due to its calculation theory. Despite the aforementioned problems, buildings with complete data are extracted successfully, as shown in Figure 13c.

4. Discussion

Point features, parameters x_0 of the logistic function to normalize each feature, weight λ_1 , λ_2 of each feature in the data term and smooth term are important to the final results. Therefore, discussions

about the proposed feature, above parameters setting and the running time of the proposed method are conducted in this section.

4.1. Discussion of f_v

In the proposed method, point feature f_v based on variance of LRSC-based normal vector (hereinafter referred to as f_{LRSC}), is used to discriminate building points from vegetation. To evaluate its performance, a comparison between f_{LRSC} and feature f_v based on PCA-based normal vector (hereinafter referred to as f_{PCA}) is conducted using Area 1 to 3 datasets. In the comparison, only f_{LRSC} or f_{PCA} is used to extract buildings in the proposed method. Quality Q (%) metric on a per-area level is used to measure the accuracy of building extraction, shown in Figure 14.

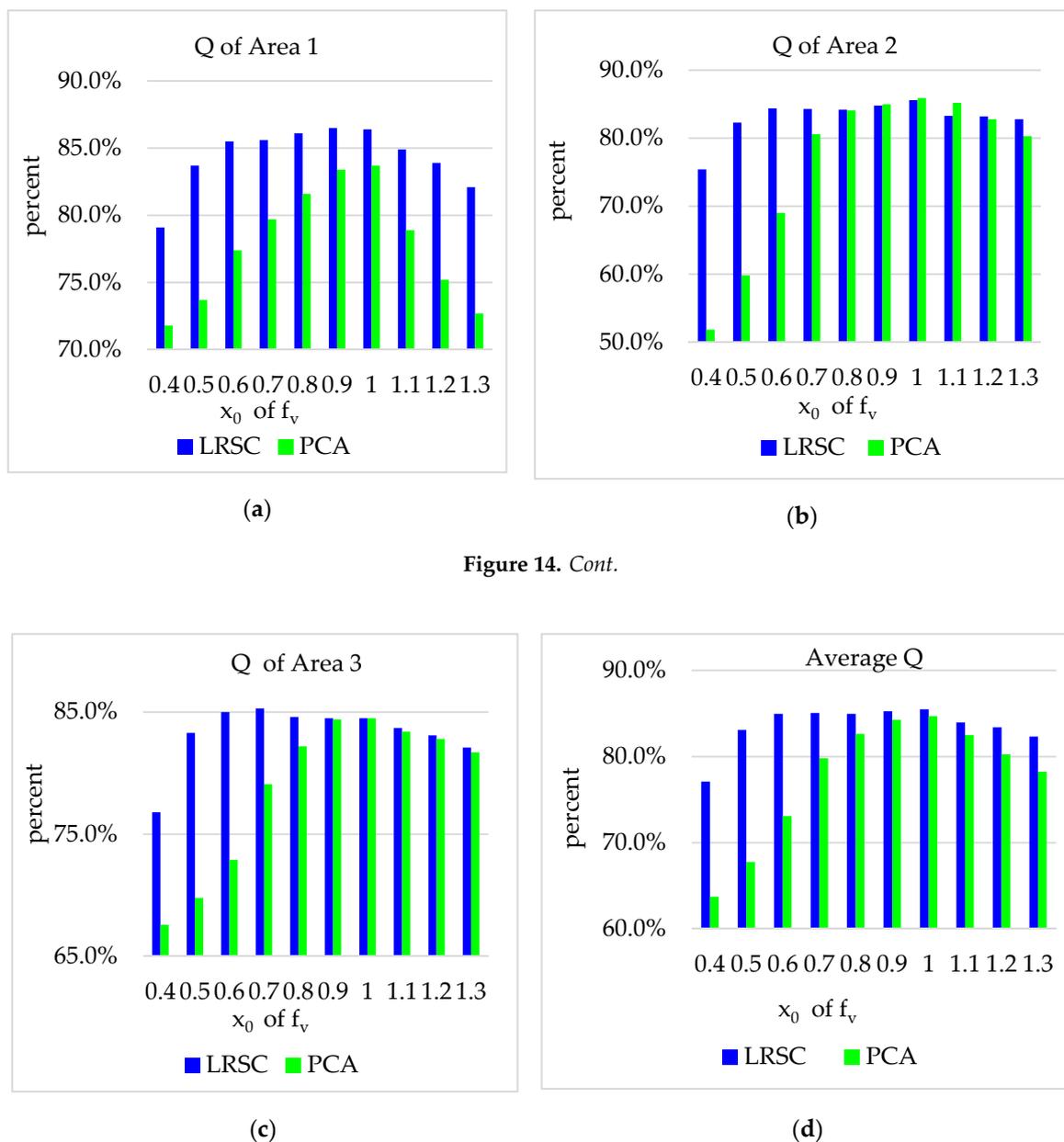


Figure 14. Cont.

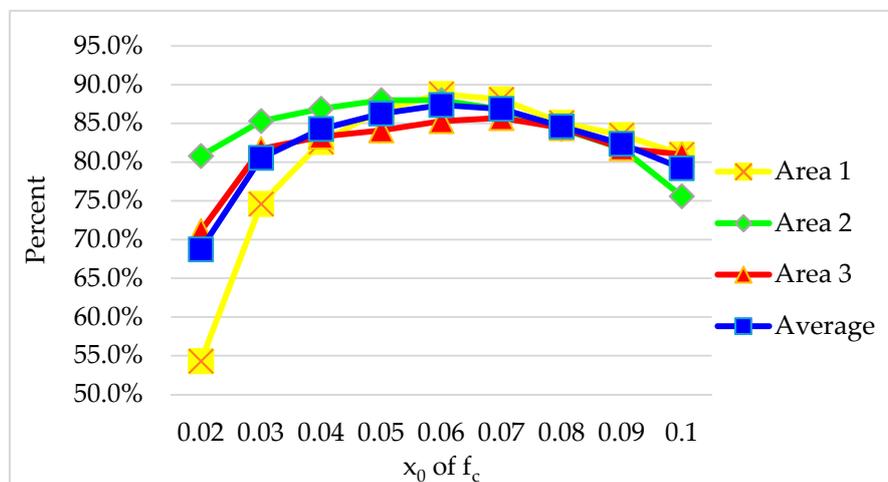
Figure 14. Q (quality) metric on a per-area level of extraction results. (a) Q of Area 1; (b) Q of Area 2; (c) Q of Area 3; (d) Average Q.

In Figure 14, the accuracy of f_{PCA} significantly changes for different x_0 compared with f_{LRSC} in Area 1 to 3, which indicates f_{PCA} is more sensitive to the feature threshold parameter x_0 . Moreover,

Q difference between two features is larger in Area 1 than in Area 2 and 3. This is because Area 1 includes rather complex historic buildings composed of irregular roof planes with different slopes, while roofs of most buildings in Area 2 are horizontal and structures of buildings in Area 3 are simple and regular. Moreover, low-rank subspace clustering (LRSC) technique can calculate accurate normal vectors compared with PCA [48]. It indicates f_{LRSC} perform much better than f_{PCA} to extract buildings in complex scenes.

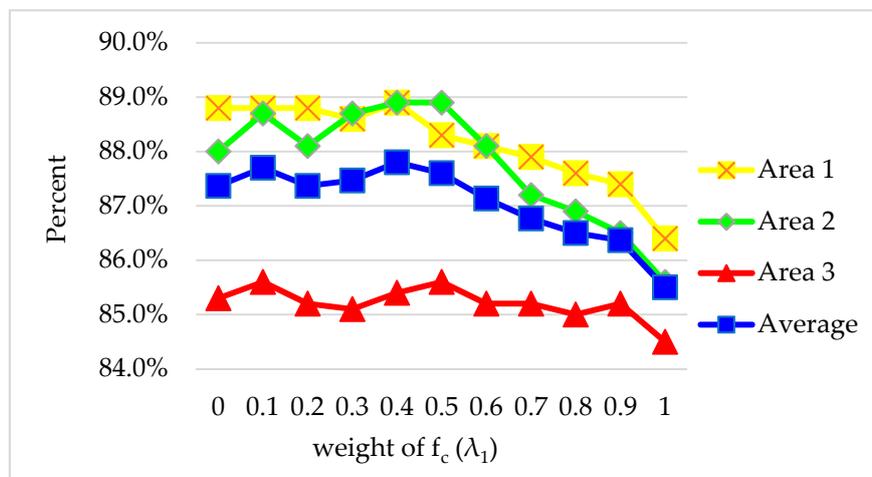
4.2. Discussion of Parameters Setting

Parameter x_0 of f_{LRSC} is set to 1.0 according to the results in Figure 14. The Q (%) on a per-area level is used to study the optimal x_0 of f_c using Area 1 to 3 datasets. Note that only f_c is used in the proposed method, and Q is shown in Figure 15a. It can be seen that Q is more sensitive to x_0 in Area 1 than in the other two areas, possibly due to the complex buildings, and Q reaches the maximum when x_0 is set to the proper value for these three areas. According to the average results, when the optimal x_0 of f_c is set to 0.06, the highest Q is obtained.



(a)

Figure 15. Cont.



(b)

Figure 15. Results of different x_0 for f_c and different weights for features. (a) Results of different x_0 for f_c ; (b) Results of different weights for each feature.

When analyzing the weight parameters λ_1 , λ_2 , one is adjusted from 0 to 1 and the other is correspondingly set as from 1 to 0. The metric Q on a per-area level is used to study the optimal λ_1 and λ_2 , and the Q is shown in Figure 15b. The highest average Q is obtained when $\lambda_1 = 0.4$ and $\lambda_2 = 0.6$, and accuracy will be improved by combining two features together.

4.3. Discussion of Running Time

Experiments of ISPRS datasets (i.e., Area 1 to 5) and other two LiDAR datasets (i.e., New Zealand dataset and Utah dataset) were performed on a laptop computer with 16 GB RAM and an Intel Core i7-7700HQ @ 2.8 GHz CPU, and a Windows 10 64-bit operating system. The proposed method was implemented using C++ with the platform of Visual Studio 2013. It should be noted that the total running time (T) of the proposed method is composed of two parts: time (T_1) before post-processing (i.e., Step 1 and 2) and time (T_2) of post-processing (i.e., Step 3), listed in Table 5.

Table 5. Running time (s).

| Item | Area ID | | | | | | |
|-------|---------|--------|--------|--------|--------|---------------------|--------------|
| | Area 1 | Area 2 | Area 3 | Area 4 | Area 5 | New Zealand Dataset | Utah Dataset |
| T_1 | 14 | 18 | 20 | 357 | 307 | 70 | 127 |
| T_2 | 23 | 51 | 61 | 4831 | 4642 | 653 | 2239 |
| T | 37 | 69 | 81 | 5188 | 4949 | 723 | 2366 |

From Table 5, it can be seen that T_1 are significantly less than T_2 due to fact that only two point features need to be calculated, and graph cuts and PTD algorithms are efficient [44,45,54]. Therefore, satisfying initial extraction results can be obtained using less time. Through analysis, step of the maximum intersection angle constraint occupies the most time in the step of post-processing. This is because large numbers of neighbors are searched to calculate angles and obtain the maximum intersection angle to eliminate non-building points of small clusters. Area-based method can be used to eliminate these non-building points with high efficiency [12], but it is sensitive to area threshold and fails to eliminate large non-building point clusters with narrow width compared with the maximum intersection angle constraint. A possible solution to the problem is the combination of area-based method and the maximum intersection angle constraint. How to combine these two methods and their potential influence to eliminate non-building points are worth further study. It should be noted that running time of Area 4 and 5 is far more than Area 1 to 3 due to the fact that number of points in Area 4 and 5 is far more than Area 1 to 3, and scenes in Area 4 to 5 are more complex.

5. Conclusions

Due to the complexity of urban scenes, it is still a challenging task to extract buildings automatically. In the paper, a building extraction method from airborne LiDAR data based on min-cut and improved post-processing is proposed. To discriminate building points on the intersecting roof planes from vegetation, a point feature based on variance of normal vectors via low-rank subspace clustering (LRSC) is proposed, and non-ground points are separated into two subsets based on min-cut after filtering. Then, an improved post-processing is adopted to refine building extraction results by using restricted region growing and the constraints of height, the maximum intersection angle, and consistency. Omitted points of buildings located in slopes are detected using the restricted region growing. The proposed maximum intersection angle constraint effectively removes large non-building point clusters with narrow width, such as greenbelt along street, overcoming the defects of area-based methods in setting area threshold. Contextual information and consistency constraint are both used to eliminate inhomogeneity in the process of building extraction. No manual operations are required in the process except predefining some threshold values. Experiments of seven datasets verify that most buildings, even with curved roofs, are successfully extracted by the proposed method. In terms of precision, for Area 1 to 3 in Vaihingen the average Q metrics of the proposed method achieves promising

results with 87.7%, 83.7%, and 99.1% for the per-area, per-object, per-object >50 m² levels respectively. And for Area 4 to 5 in Toronto, the average Q metrics are 88.9% for the per-area level. The Q metric for the per-area level of the proposed method achieves 93.2% for high-density dataset with average point density 20 points/m². Moreover, the proposed point feature outperforms the comparison alternative and is less sensitive to feature threshold in complex scenes. In addition, only two point features are used in the proposed method, which beneficially decreases the computation cost of calculating point features and improving algorithmic efficiency.

However, some defects still exist in the proposed method. It is still a challenge for the proposed method to successfully extract some buildings attached by complex skylight or roof terrace due to rough points. A feasible solution is to combine images or intensity data to obtain extra features [6,10], which deserves further studies. In addition, there are several parameters that are adopted in the proposed method, which reduces the full automation of building extraction. We will attempt to construct a self-adaptive building extraction algorithm in the next step. Moreover, there are other normalization functions available, such as min-max normalization, Z-score normalization, etc. [66], but only logical function is employed to normalize features. How to introduce other normalization functions to normalize feature and their potential influence to building extraction are worth further research.

Author Contributions: Conceptualization, K.L.; methodology, K.L.; software, L.Z. and H.M. (Haichi Ma); validation, H.M. (Hongchao Ma) and Z.C.; formal analysis, K.L. and H.M. (Hongchao Ma); investigation, K.L.; resources, H.M. (Haichi Ma) and K.L.; data curation, H.M. (Haichi Ma) and K.L.; writing—original draft preparation, K.L.; writing—review and editing, H.M. (Hongchao Ma) and Z.C.; visualization, K.L. and H.M. (Haichi Ma); supervision, H.M. (Hongchao Ma); project administration, H.M. (Hongchao Ma); funding acquisition, H.M. (Hongchao Ma) and L.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Key R&D Program of China (2018YFB0504500), National Natural Science Foundation of China (Grant numbers 41601504 and 61378078), and National High Resolution Earth Observation Foundation (11-Y20A12-9001-17/18).

Acknowledgments: The test datasets were provided by the International Society for Photogrammetry and Remote Sensing (ISPRS) Working Group and OpenTopography. Authors would like to thank them and anonymous reviewers for their constructive comments for the manuscript.

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

References

1. Wang, R.; Peethambaran, J.; Chen, D. LiDAR point clouds to 3-D Urban Models: A review. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2018**, *11*, 606–627. [[CrossRef](#)]
2. Guo, L.; Luo, J.; Yuan, M.; Huang, Y.; Shen, H.; Li, T. The influence of urban planning factors on PM2.5 pollution exposure and implications: A case study in China based on remote sensing, LBS, and GIS data. *Sci. Total Environ.* **2019**, *659*, 1585–1596. [[CrossRef](#)] [[PubMed](#)]
3. Janalipour, M.; Mohammadzadeh, A. A novel and automatic framework for producing building damage map using post-event LiDAR data. *Int. J. Disaster Risk Reduct.* **2019**, *39*, 1–13. [[CrossRef](#)]
4. Peng, Z.; Gao, S.; Xiao, B.; Guo, S.; Yang, Y. CrowdGIS: Updating digital maps via mobile crowdsensing. *IEEE Trans. Autom. Sci. Eng.* **2017**, *15*, 369–380. [[CrossRef](#)]
5. Zhou, Z.; Gong, J. Automated residential building detection from airborne LiDAR data with deep neural networks. *Adv. Eng. Inform.* **2018**, *36*, 229–241. [[CrossRef](#)]
6. Chen, S.; Shi, W.; Zhou, M.; Zhang, M.; Chen, P. Automatic building extraction via adaptive iterative segmentation with LiDAR data and high spatial resolution imagery fusion. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2020**, *13*, 2081–2095. [[CrossRef](#)]
7. Mallet, C.; Bretar, F.; Roux, M.; Soergel, U.; Heipke, C. Relevance assessment of full-waveform lidar data for urban area classification. *ISPRS J. Photogramm. Remote Sens.* **2011**, *66*, S71–S84. [[CrossRef](#)]
8. Donoghue, D.N.M.; Watt, P.J.; Cox, N.J.; Wilson, J. Remote sensing of species mixtures in conifer plantations using LiDAR height and intensity data. *Remote Sens.* **2007**, *110*, 509–522. [[CrossRef](#)]

9. Salimzadeh, N.; Hammad, A. High-level framework for GIS-based optimization of building photovoltaic potential at urban scale using BIM and LiDAR. In Proceedings of the International Conference on Sustainable Infrastructure, New York, NY, USA, 26–28 October 2017; pp. 123–134. [\[CrossRef\]](#)
10. Awrangjeb, M.; Zhang, C.; Fraser, C. Automatic extraction of building roofs using LIDAR data and multispectral imagery. *ISPRS J. Photogramm. Remote Sens.* **2013**, *83*, 1–18. [\[CrossRef\]](#)
11. Awrangjeb, M.; Fraser, C. Automatic segmentation of raw LiDAR data for extraction of building roofs. *Remote Sens.* **2014**, *6*, 3716–3751. [\[CrossRef\]](#)
12. Du, S.; Zhang, Y.; Zou, Z.; Xu, S.; He, X.; Chen, S. Automatic building extraction from LiDAR data fusion of point and grid-based features. *ISPRS J. Photogramm. Remote Sens.* **2017**, *130*, 294–307. [\[CrossRef\]](#)
13. Ni, H.; Lin, X.; Zhang, J. Classification of ALS point cloud with improved point cloud segmentation and random forests. *Remote Sens.* **2017**, *9*, 288. [\[CrossRef\]](#)
14. Vosselman, G.; Coenen, M.; Rottensteiner, F. Contextual segment-based classification of airborne laser scanner data. *ISPRS J. Photogramm. Remote Sens.* **2017**, *128*, 354–371. [\[CrossRef\]](#)
15. Guo, B.; Huang, X.; Zhang, F.; Sohn, G. Classification of airborne laser scanning data using JointBoost. *ISPRS J. Photogramm. Remote Sens.* **2015**, *100*, 71–83. [\[CrossRef\]](#)
16. Zhang, J.; Lin, X.; Ning, X. SVM-based classification of segmented airborne LiDAR point clouds in urban areas. *Remote Sens.* **2013**, *5*, 3749–3775. [\[CrossRef\]](#)
17. Breiman, L. Random forests. *Mach. Learn.* **2001**, *45*, 5–32. [\[CrossRef\]](#)
18. Moon, T.K. The expectation-maximization algorithm. *IEEE Signal. Process. Mag.* **1996**, *13*, 47–60. [\[CrossRef\]](#)
19. Torlay, L.; Perrone-Bertolotti, M.; Thomas, E.; Baciú, M. Machine learning–XGBoost analysis of language networks to classify patients with epilepsy. *Brain Inform.* **2017**, *4*, 1–11. [\[CrossRef\]](#)
20. Zhang, Z.; Zhang, L.; Tong, X.; Mathiopoulos, T.; Guo, B.; Huang, X.; Wang, Z.; Wang, Y. A multilevel point-cluster-based discriminative feature for ALS point cloud classification. *IEEE Trans. Geosci. Remote Sens.* **2016**, *54*, 3309–3321. [\[CrossRef\]](#)
21. Weinmann, M.; Jutzi, B.; Mallet, C. Semantic 3D scene interpretation: A framework combining optimal neighborhood size selection with relevant features. *ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci.* **2014**, *2*, 181–188. [\[CrossRef\]](#)
22. Wang, C.; Shu, Q.; Wang, X.; Guo, B.; Liu, P.; Li, Q. A random forest classifier based on pixel comparison features for urban LiDAR data. *ISPRS J. Photogramm. Remote Sens.* **2019**, *148*, 75–86. [\[CrossRef\]](#)
23. Huang, R.; Yang, B.; Liang, F.; Dai, W.; Li, J.; Tian, M.; Xu, W. A top-down strategy for buildings extraction from complex urban scenes using airborne LiDAR point clouds. *Infrared Phys. Technol.* **2018**, *92*, 203–218. [\[CrossRef\]](#)
24. Wang, Y.; Jiang, T.; Yu, M.; Tao, S.; Sun, J.; Liu, S. Semantic-based building extraction from LiDAR point clouds using contexts and optimization in complex environment. *Sensors* **2020**, *20*, 3386. [\[CrossRef\]](#) [\[PubMed\]](#)
25. Dong, W.; Lan, J.; Liang, S.; Yao, W.; Zhang, Z. Selection of LiDAR geometric features with adaptive neighborhood size for urban land cover classification. *Int. J. Appl. Earth Obs. Geoinf.* **2017**, *60*, 99–110. [\[CrossRef\]](#)
26. Han, W.; Wang, R.; Huang, D.; Xu, C. Large-Scale ALS data semantic classification integrating location-context-semantics cues by higher-order CRF. *Sensors* **2020**, *20*, 1700. [\[CrossRef\]](#)
27. Cai, Z.; Ma, H.; Zhang, L. Feature selection for airborne LiDAR data filtering: A mutual information method with Parzon window optimization. *Glsci. Remote Sens.* **2020**, *57*, 323–337. [\[CrossRef\]](#)
28. 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. [\[CrossRef\]](#)
29. 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. [\[CrossRef\]](#)
30. Awrangjeb, M.; Lu, G.; Fraser, C. Automatic building extraction from LiDAR data covering complex urban scenes. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2014**, *40*, 25–32. [\[CrossRef\]](#)
31. Wang, Y.; Sun, Y.; Liu, Z.; Sarma, S.E.; Bronstein, M.M.; Solomon, J.M. Dynamic graph cnn for learning on point clouds. *ACM Trans. Graph.* **2019**, *38*, 1–12. [\[CrossRef\]](#)
32. Hu, X.; Yuan, Y. Deep-Learning-Based classification for DTM extraction from ALS point cloud. *Remote Sens.* **2016**, *8*, 730. [\[CrossRef\]](#)
33. Yao, X.; Guo, J.; Hu, J.; Cao, Q. Using deep learning in semantic classification for point cloud data. *IEEE Access* **2019**, *7*, 37121–37130. [\[CrossRef\]](#)

34. Verma, V.; Kumar, R.; Hsu, S. 3D building detection and modeling from aerial LiDAR data. In Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, IEEE Computer Society, Washington, DC, USA, 17–22 June 2006; pp. 2213–2220. [[CrossRef](#)]
35. 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. [[CrossRef](#)]
36. Borrmann, D.; Elseberg, J.; Lingemann, K.; Nüchte, A. The 3d hough transform for plane detection in point clouds: A review and a new accumulator design. *3D Res.* **2011**, *2*, 1–13. [[CrossRef](#)]
37. Cai, Z.; Ma, H.; Zhang, L. A Building detection method based on semi-suppressed fuzzy C-means and restricted region growing using airborne LiDAR. *Remote Sens.* **2019**, *11*, 848. [[CrossRef](#)]
38. Adhikari, S.K.; Sing, J.K.; Basu, D.K.; Nasipuri, M. Conditional spatial fuzzy C-means clustering algorithm for segmentation of MRI images. *Appl. Soft Comput.* **2015**, *34*, 758–769. [[CrossRef](#)]
39. Meng, X.; Wang, L.; Currit, N. Morphology-Based building detection from airborne LIDAR data. *Photogramm. Eng. Remote Sens.* **2009**, *75*, 437–442. [[CrossRef](#)]
40. Cheng, L.; Zhao, W.; Han, P.; Zhang, W.; Shan, J.; Liu, Y. Building region derivation from LiDAR data using a reversed iterative mathematic morphological algorithm. *Opt. Commun.* **2013**, *286*, 244–250. [[CrossRef](#)]
41. Gerke, M.; Xiao, J. Fusion of airborne laserscanning point clouds and images for supervised and unsupervised scene classification. *ISPRS J. Photogramm. Remote Sens.* **2014**, *87*, 78–92. [[CrossRef](#)]
42. Rusu, R.B.; Cousins, S. 3D is here: Point cloud library. In Proceeding of the IEEE International Conference on Robotics and Automation, Shanghai, China, 9–13 May 2011; pp. 1–4. [[CrossRef](#)]
43. Rusu, R.B.; Marton, Z.C.; Blodow, N.; Dolha, M.; Beetz, M. Towards 3D point cloud based object maps for household environments. *Rob. Auton. Syst.* **2008**, *56*, 927–941. [[CrossRef](#)]
44. Ma, H.; Zhou, W.; Zhang, L. DEM refinement by low vegetation removal based on the combination of fullwaveform data and progressive TIN densification. *ISPRS J. Photogramm. Remote Sens.* **2018**, *146*, 260–271. [[CrossRef](#)]
45. Axelsson, P. Processing of laser scanner data—algorithms and applications. *ISPRS J. Photogramm. Remote Sens.* **1999**, *54*, 138–147. [[CrossRef](#)]
46. Liu, K.; Ma, H.; Zhang, L.; Cai, Z.; Ma, H. Strip adjustment of airborne LiDAR data in urban scenes using planar features by the minimum hausdorff distance. *Sensors* **2019**, *19*, 5131. [[CrossRef](#)] [[PubMed](#)]
47. Blomley, R.; Jutzi, B.; Weinmann, M. Classification of airborne laser scanning data using geometric, multi-scale features and different neighborhood types. *ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci.* **2016**, *3*, 169–176. [[CrossRef](#)]
48. Zhang, J.; Cao, J.; Liu, X.; Wang, J.; Liu, J.; Shi, X. Point cloud normal estimation via low-rank subspace clustering. *Comput. Graph.* **2013**, *37*, 697–706. [[CrossRef](#)]
49. ISPRS. Available online: <http://www2.isprs.org/commissions/comm3/wg4/tests.html> (accessed on 22 August 2020).
50. Li, Y.; Tong, G.; Du, X.; Yang, X.; Zhang, J.; Yang, L. A single point-based multilevel features fusion and pyramid neighborhood optimization method for ALS point cloud classification. *Appl. Sci.* **2019**, *9*, 951. [[CrossRef](#)]
51. 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](#)]
52. Delong, A.; Osokin, A.; Isack, H.N.; Boykov, Y. Fast approximate energy minimization with label costs. *Int. J. Comput. Vis.* **2012**, *96*, 1–27. [[CrossRef](#)]
53. Ural, S.; Shan, J. Min-Cut based segmentation of airborne LiDAR point clouds. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2012**, *39*, 167–172. [[CrossRef](#)]
54. Boykov, Y.; Kolmogorov, V. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE Trans. Pattern Anal. Mach. Intell.* **2004**, *26*, 1124–1137. [[CrossRef](#)]
55. Li, W.; Wang, Y.; Du, J.; Lai, J. Synergistic integration of graph-cut and cloud model strategies for image segmentation. *Neurocomputing* **2017**, *257*, 37–46. [[CrossRef](#)]
56. Guo, Y.; Akbulut, Y.; Şengür, A.; Xia, R.; Smarandache, F. An efficient image segmentation algorithm using neutrosophic graph cut. *Symmetry* **2017**, *9*, 185. [[CrossRef](#)]
57. Reza, M.N.; Na, I.S.; Baek, S.W.; Lee, K.H. Rice yield estimation based on K-means clustering with graph-cut segmentation using low-altitude UAV images. *Biosyst. Eng.* **2019**, *177*, 109–121. [[CrossRef](#)]

58. Sánchez-Lopera, J.; Lerma, J.L. Classification of lidar bare-earth points, buildings, vegetation, and small objects based on region growing and angular classifier. *Int. J. Remote Sens.* **2014**, *35*, 6955–6972. [[CrossRef](#)]
59. Truong, L.T.; Nguyen, H.T.; Nguyen, H.D.; Vu, H.V. Pedestrian overpass use and its relationships with digital and social distractions, and overpass characteristics. *Accid. Anal. Prev.* **2019**, *131*, 234–238. [[CrossRef](#)]
60. OpenTopography. Available online: <https://portal.opentopography.org/datasetMetadata?otCollectionID=OT.022020.2193.2> (accessed on 22 August 2020). [[CrossRef](#)]
61. OpenTopography. Available online: <https://portal.opentopography.org/datasetMetadata?otCollectionID=OT.122014.26912.1> (accessed on 22 August 2020). [[CrossRef](#)]
62. Ardila, J.P.; Tolpekin, V.A.; Bijker, W.; Stein, A. Markov-Random-Field-Based super-resolution mapping for identification of urban trees in VHR images. *ISPRS J. Photogramm. Remote Sens.* **2011**, *66*, 762–775. [[CrossRef](#)]
63. Rutzinger, M.; Rottensteiner, F.; Pfeifer, N. A comparison of evaluation techniques for building extraction from airborne laser scanning. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2009**, *2*, 11–20. [[CrossRef](#)]
64. Ma, L.; Li, Y.; Li, J.; Zhong, Z.; Chapman, M. Generation of horizontally curved driving lines in HD maps using mobile laser scanning point clouds. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2019**, *12*, 1572–1586. [[CrossRef](#)]
65. Powers, D. Evaluation: From precision, recall and F-measure to ROC, informedness, markedness and correlation. *J. Mach. Learn. Tech.* **2011**, *2*, 37–63.
66. Eesa, A.S.; Arabo, W.K. A normalization methods for backpropagation: A comparative study. *Sci. J. Univ. Zakho* **2017**, *5*, 319–323. [[CrossRef](#)]



© 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/>).