

Article

## Automatic Segmentation of Raw LIDAR Data for Extraction of Building Roofs

Mohammad Awrangjeb <sup>1,\*</sup> and Clive S. Fraser <sup>2</sup>

<sup>1</sup> Gippsland School of Information Technology, Monash University, Melbourne, VIC 3842, Australia

<sup>2</sup> CRC for Spatial Information, Department of Infrastructure Engineering, University of Melbourne, Melbourne, VIC 3010, Australia; E-Mail: c.fraser@unimelb.edu.au

\* Author to whom correspondence should be addressed; E-Mail: mohammad.awrangjeb@monash.edu; Tel.: +61-3-9902-6462; Fax: +61-3-9902-7137.

Received: 15 January 2014; in revised form: 3 April 2014 / Accepted: 9 April 2014 /

Published: 28 April 2014

---

**Abstract:** Automatic extraction of building roofs from remote sensing data is important for many applications, including 3D city modeling. This paper proposes a new method for automatic segmentation of raw LIDAR (light detection and ranging) data. Using the ground height from a DEM (digital elevation model), the raw LIDAR points are separated into two groups. The first group contains the ground points that form a “building mask”. The second group contains non-ground points that are clustered using the building mask. A cluster of points usually represents an individual building or tree. During segmentation, the planar roof segments are extracted from each cluster of points and refined using rules, such as the coplanarity of points and their locality. Planes on trees are removed using information, such as area and point height difference. Experimental results on nine areas of six different data sets show that the proposed method can successfully remove vegetation and, so, offers a high success rate for building detection (about 90% correctness and completeness) and roof plane extraction (about 80% correctness and completeness), when LIDAR point density is as low as four points/m<sup>2</sup>. Thus, the proposed method can be exploited in various applications.

**Keywords:** LIDAR; point cloud; segmentation; automatic; building; roof; extraction

---

## 1. Introduction

Automatic extraction of buildings from aerial imagery and/or LIDAR (light detection and ranging) data is a prerequisite for many GIS (Geographic Information System) applications, such as 3D building modelling [1]. Building extraction implies the extraction of 2D or 3D building information, such as individual building and roof plane boundaries. Although the problem is well understood and, in many cases, accurate results are delivered, the major drawback is that the current level of automation in building extraction is comparatively low [2].

Based on the usage of the input data (imagery, LIDAR data, *etc.*), there are three main categories of building extraction methods. The first category [3] fully relies on high-resolution aerial imagery. Although promising results have been shown, the imagery-only approach does not generally perform well in densely built-up areas, partially due to shadows, occlusions and poor contrast. Consequently, automated building extraction from aerial imagery alone is generally not reliable enough for practical implementation [4]. The second category of methods employs LIDAR data and offers an improved level of automation when compared to image-only methods [5,6]. Methods in the third category integrate aerial imagery and LIDAR data to exploit the complementary information from both data sources [7].

Methods integrating LIDAR data and imagery can face one or more of the following difficulties: First, accurately coregistered imagery and point cloud data is a prerequisite. However, good quality registered data may not be available, and the registration of two data sources having dissimilar characteristics can be a difficult task. Second, there can be changes in the scene when the point cloud and imagery are captured on different, well-separated dates. Third, in a highly vegetated area, there may be dense vegetation close to a building. Thus, the required building information in the scene may not be available due to shadows and occlusions. Fourth, the imagery that is nowadays available usually has a high spatial resolution, such that the level of detail in a scene can be too high for building extraction purposes. For example, small structures (chimneys, gutters, *etc.*) might be well recognised in an image. Therefore, the separation of relevant information from that which is irrelevant is difficult. As a consequence, practical implementations of such building extraction methods tend to be semiautomatic processes, as exemplified by the method reported in Khoshelham *et al.* [8]. Finally, when a true orthoimage is not available, artifacts in the orthoimage can provide misleading information. For example, the recently developed method by Awrangjeb *et al.* [7] can fail to extract a roof plane when the corresponding roof edge or ridge line cannot be extracted from the orthoimage. Thus, the integration of aerial imagery and LIDAR data for building extraction remains a difficult task, due to the dissimilar characteristics of the two data sources.

Recently, digital surface models (DSMs) generated via dense image matching from highly overlapping aerial images [9] have been used as an alternative to LIDAR point clouds for building extraction [10]. Since these DSMs are usually denser than those from LIDAR, they can be used to enhance feature extraction [9].

The study reported in this paper utilises only LIDAR data for building roof extraction. LIDAR-only methods fall into two main categories, depending upon the output information (building and plane boundaries). The methods in the first category detect building boundaries or footprints [11]. The methods in the second category extract boundaries of individual roof planes [12]. Some in the latter category

also reconstruct individual building roof models by regularising the plane boundaries and defining the relationships among the extracted roof planes [13].

This paper proposes an automatic technique for the extraction of building and roof plane boundaries using LIDAR point cloud data. However, it does not consider the subsequent generation of individual building roof models. In the proposed method, the LIDAR points are first divided into ground and non-ground points. The ground points are used to generate a building mask. The black shapes (voids) formed in the mask represent the ground areas covered by buildings and trees, where there are no laser returns below a given height. Therefore, pixels corresponding to individual buildings and trees are clustered in the building mask. The non-ground LIDAR points within each cluster are then segmented based on their coplanarity properties and neighbourhood relationships. A new procedure is applied to the extracted planar segments in order to reduce the problems of under- and over-segmentation. Another procedure is applied to remove false planes, mainly constructed in trees. These “tree planes” are usually small in size and randomly oriented. They can be removed easily using information, such as area, long straight line segments along the plane boundary and neighbourhood relationships, as well as via out-of-plane point “spikes” within the planar boundary. Finally, the neighbouring planes are grouped in order to obtain the boundary of an individual building. Experimental results show that the proposed technique can provide good building detection and roof plane extraction results.

An initial version of the proposed method was published in [14]. Here, the proposed method is presented in more detail. The selection of values of input parameters for the proposed algorithm are analysed, and the results obtained from six data sets are presented. Experimental results are compared with those achieved via five existing methods. The proposed method has, in two aspects, some similarity with the methods reported in [7,15], since a building mask is initially generated and the LIDAR points are divided into two groups (ground and non-ground) following the procedure adopted in [7,16]. However, unlike the method reported in Awrangjeb *et al.* [7], the proposed method does not use aerial imagery to identify the building regions. Instead, points in the building mask are clustered to obtain the non-ground objects, which are mostly buildings, but may include some dense vegetation. The non-ground points within each cluster are further divided into coplanar and non-coplanar points. The coplanarity of a point with respect to its neighbours is quantified using an eigenvalue analysis, similar to the approach adopted in [15]. However, as the local surface normals are quite noisy in dense point clouds [17], a down-sampling of the point cloud is first carried out in order to determine whether the majority of the sampled points on a nominally planar surface are indeed coplanar. Other points around each of the sampled points are included during the extraction of planar segments through the use of a new region growing algorithm that exploits the coplanar points in order to define stable seed regions. New procedures are also proposed for the refinement of the extracted building roof segments, in order to both reduce the effect of over- and under-segmentation and to remove tree segments.

The rest of the paper is organised as follows: Section 2 presents a review of alternative methods based on LIDAR data that have been reported in the literature for building and roof extraction. Section 3 details the proposed building roof extraction algorithm. Section 4 presents experimental results for nine test areas from six data sets. The selection of values for algorithmic parameters and a comparison of results from the proposed technique with those achieved via existing methods are also discussed in Section 4. Concluding remarks are then provided in Section 5.

## 2. Related Work

Recent research into building extraction from raw LIDAR data can be divided into two major categories [18]. The first tries to fit certain shapes to the data, while the second tries to extract shapes present in the data. Methods in the first type are thus model-driven, since they require a predefined catalogue of building models. Although they are robust, their performance is limited to the known models. Methods in the second type are data-driven and, therefore, work in general for any building shape. In this section, a review of selected recent data-driven methods is presented.

Zhang *et al.* [11] employed the separation of ground and non-ground points using a progressive morphological filter. Then, buildings were separated from trees by applying a region-growing algorithm based on a plane-fitting technique. Finally, a raw building boundary was extracted by connecting boundary points, and this was regularised to establish a building footprint. The area omission and commission errors were about 12% for both residential and industrial buildings. Miliaris and Kokkas [19] used a raster DSM, generated from LIDAR data, for the extraction of building classes using region-growing-based segmentation and object-based classification algorithms. A major drawback with this method was that it required user interaction to a certain level in order to set the algorithmic parameters appropriately for different situations. Liu *et al.* [20] employed a neural network approach to extract buildings from a DSM derived from LIDAR data. The grey level co-occurrence matrix and height texture were used for the separation of buildings and trees.

Unlike Zhang *et al.* [11] and Miliaris and Kokkas [19], who used a fixed window size for filters, Cheng *et al.* [21] progressively changed the window size for morphological operations and achieved 5% to 9% omission and commission errors. Nevertheless, the use of such an adaptive algorithm to select a window size can be computationally expensive. dos Santos Galvanin and Poz [22] first segmented a LIDAR DSM using a recursive splitting technique followed by region-growing in order to identify buildings and trees. They then distinguished buildings from trees by optimising an energy function using a simulated annealing algorithm. However, this method removed some buildings along with the trees. Zhou and Neumann [23] automatically determined the principal directions of building boundaries and used them to produce footprints. However, this method worked only for flat roofs and required user interaction for the identification of non-flat roofs.

Among the methods employing individual roof plane extraction and/or building roof reconstruction, Jochem *et al.* [12] proposed a roof plane segmentation technique from raster LIDAR data using a seed point-based region growing technique. Perera *et al.* [24] used the surface growing algorithm in [25] for the segmentation of the point cloud. A cycle graph was then employed to establish the topological relationship among the line segments extracted along the plane boundaries. This method failed in the absence of missing boundary lines, and it displayed low geometric accuracy. Dorninger and Pfeifer [26] proposed a comprehensive method for the extraction, reconstruction and regularization of roof planes from LIDAR point clouds. This method involved a coarse selection of building regions by digitizing each building interactively, with erroneous building models being indicated and rectified by means of commercial CAD software. Moreover, some of the algorithmic parameters were set interactively. Sampath and Shan [15] presented a solution framework for segmentation (detection) and reconstruction of polyhedral building roofs from high-density LIDAR data. Similar to the method in [27], the

coplanarity of points was determined based on eigenvalue analysis using the Voronoi neighbourhood around each point. Kim and Shan [28] also segmented the normal vectors, but they applied a multiphase level set technique. Sun and Salvaggio [29] proposed an automated method to generate 2.5D watertight building models from LIDAR point clouds. Only visual results were presented, so there was no objective evaluation of this method. Oude Elberink and Vosselman [30] proposed a target based graph matching approach that can handle both complete and incomplete laser data. Extensive and promising experimental results were shown on four data sets.

Tarsha-Kurdi *et al.* [31] applied an extended robust estimation technique to the regenerated LIDAR point cloud. After converting the original point cloud into a DSM, the missing points were estimated as the mean of the neighbouring points. Then, a low-pass filter was applied. As a result, the regenerated points suffered from decreased positional accuracy. Moreover, the method could not construct planes with areas of less than 50 m<sup>2</sup>. Sohn *et al.* [13] clustered building points based first on height similarity and then on planar similarity. Rectilinear lines were then extracted, and polyhedral models were generated from the lines using a binary space partitioning tree. The method produced erroneous results, due to either improper localisation of the extracted lines or to missing lines. In addition, it failed to separate small roof planes in the clustering algorithm due to the use of a predefined bin size for the height histogram.

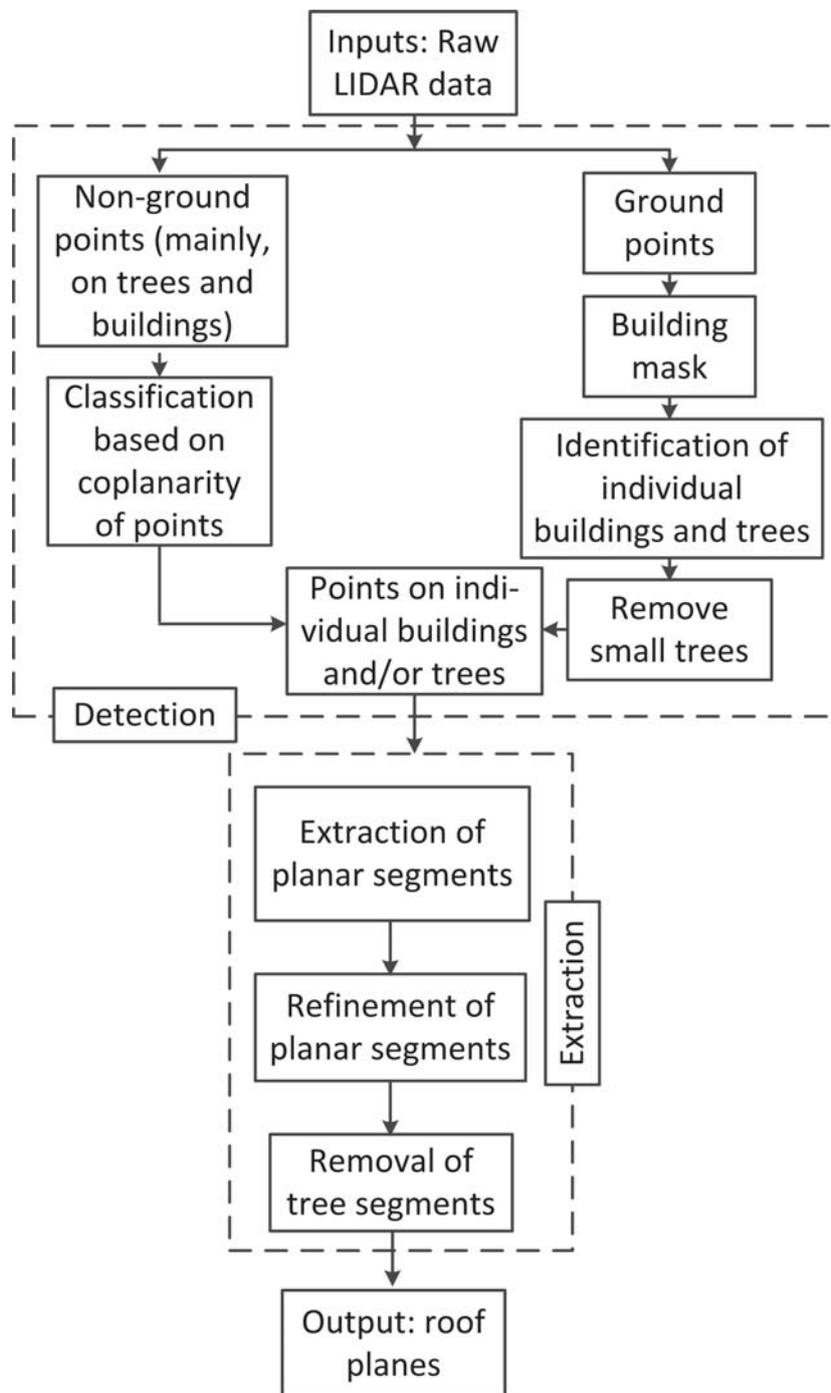
There are also methods that classify buildings separately, along with many other objects, such as ground, water, vegetation and roads. Richter *et al.* [32] segmented massive 3D point clouds using an iterative multi-pass processing scheme, in which each pass focused on different topological features and learned from already classified objects in the previous passes. Carlberg *et al.* [33] organised a cascade of binary classifiers that first identified water using a region growing segmentation algorithm and then applied 3D shape analysis for the classification of ground, buildings and trees. While this method used training data sets for object classification, that proposed by Richter *et al.* [32] did not. The use of training data sets can make a method susceptible to an unknown environment and, thus, less robust. Zhou and Neumann [34] adopted an energy minimization scheme based on the characteristic that the interior of buildings is invisible to laser scans, while trees do not possess such a characteristic, and the LIDAR points were classified into buildings, trees and ground. Niemeyer *et al.* [35] used conditional random fields in order to incorporate context knowledge from point clouds generated from full waveform LIDAR data.

### 3. Proposed Automatic Extraction

Figure 1 shows an overview of the proposed building roof extraction procedure. The input data consists of a LIDAR point cloud. In the detection step (top dashed rectangle in Figure 1), the LIDAR points are classified into two groups: ground points, such as ground, road furniture and bushes that are below a height threshold, and non-ground points, which represent elevated objects (such as buildings and trees) above this threshold. The building mask, known as the “ground mask”, is generated using the ground points. Individual buildings and trees are obtained as clusters of black pixels in the building mask, and trees with low density canopies are removed. The coplanarity of each individual non-ground LIDAR point with its neighbours is ascertained based on its Delaunay neighbourhood. The planar segments are

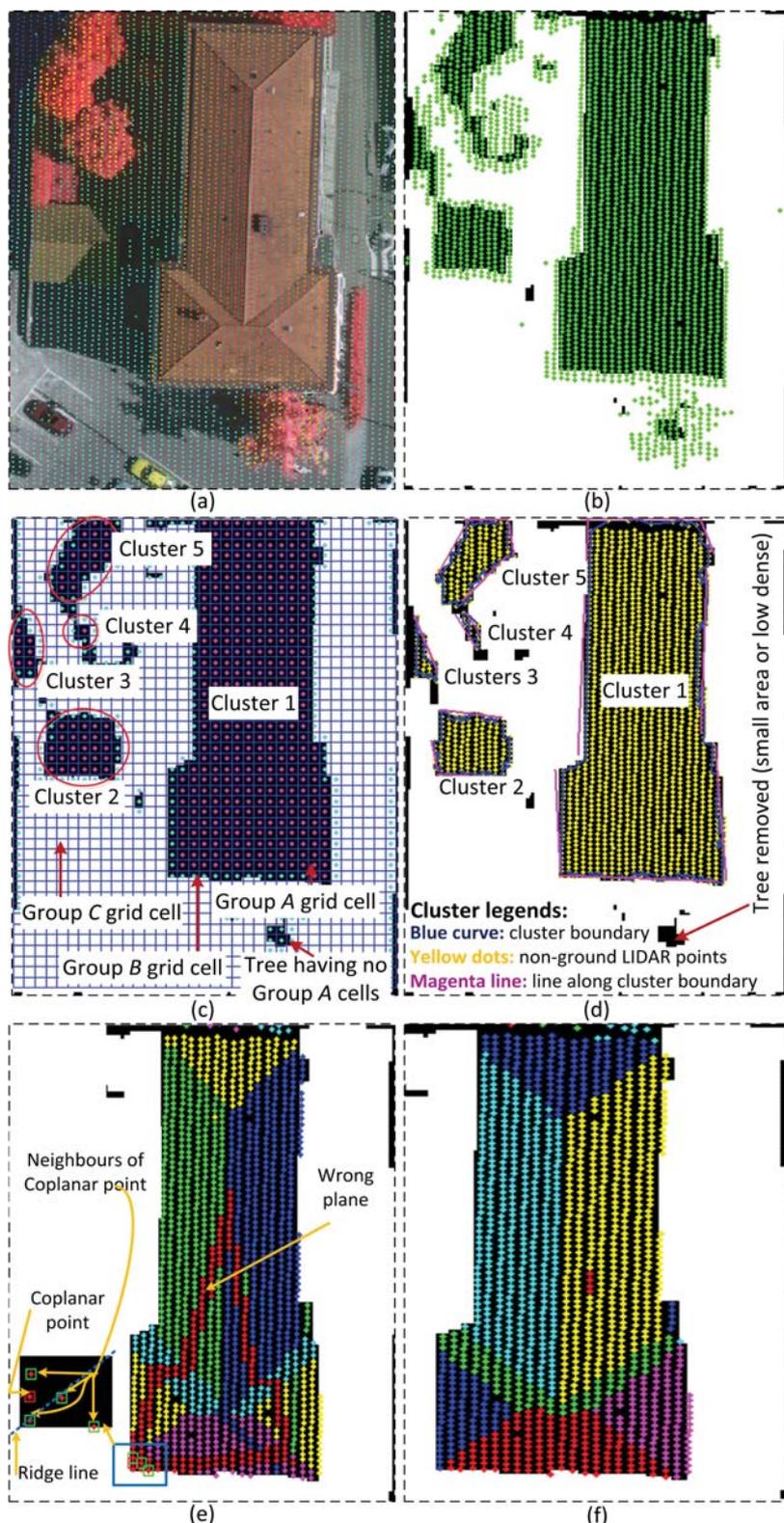
extracted from the non-ground LIDAR points on individual buildings and trees. The extracted LIDAR segments are then refined using a newly proposed procedure. Finally, the false planes on trees are removed using information, such as area, and neighbourhood characteristics, such as the presence of out-of-plane points within the planar boundary.

**Figure 1.** The proposed roof plane extraction technique.

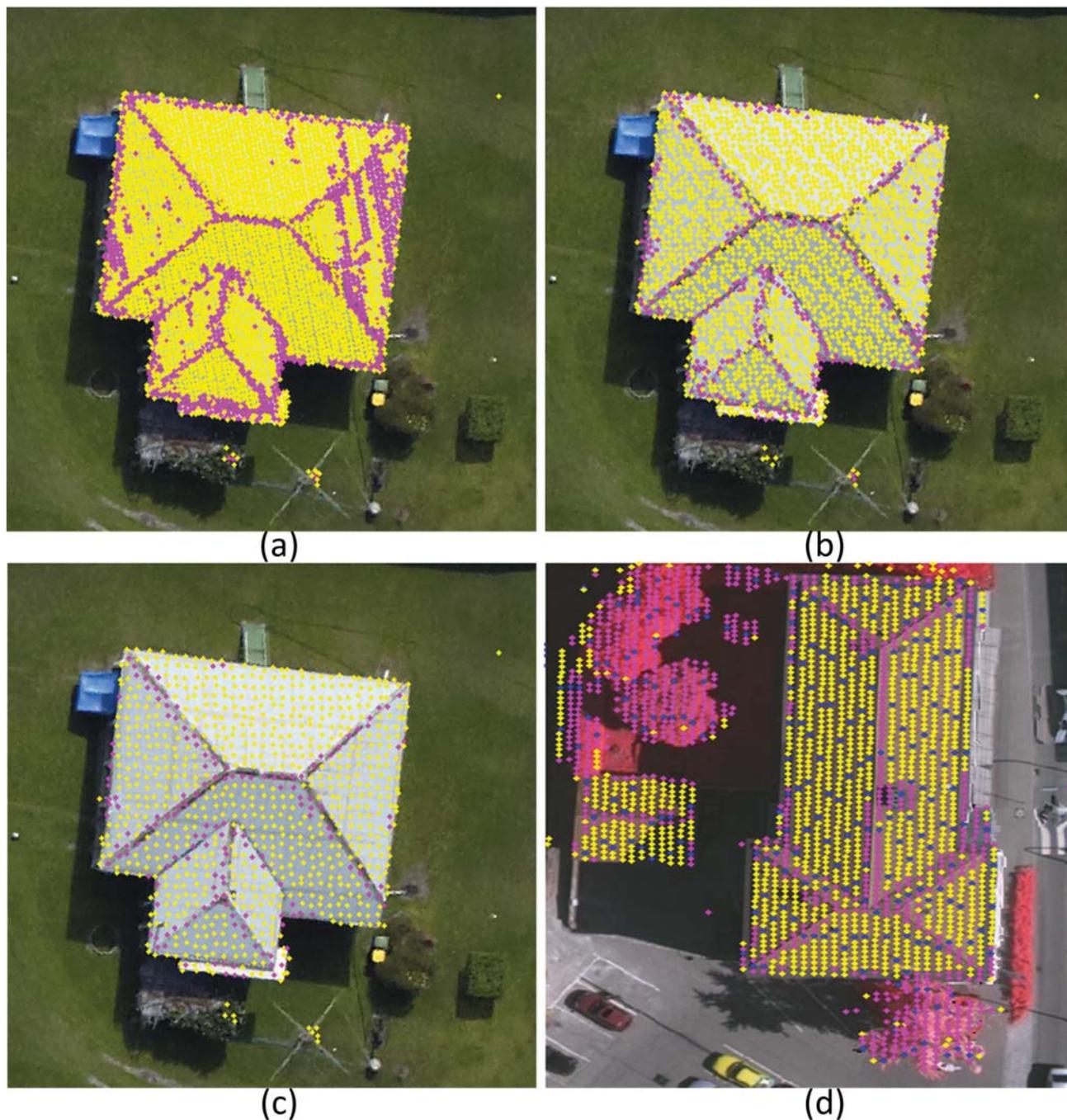


Figures 2a and 3a present two sample scenes from the Vaihingen [36] and Aitkenvale [7] data sets, respectively. They will be used to illustrate the different steps of the proposed building extraction method. The Vaihingen scene has a point density of four points/m<sup>2</sup>, whereas the Aitkenvale scene has a much higher point density of 40 points/m<sup>2</sup>.

**Figure 2.** A sample scene from the Vaihingen data set: (a) raw LIDAR points overlaid on an orthoimage (points at similar heights have similar colours); (b) building mask and non-ground LIDAR points; (c) grouping and clustering of the mask grid cells; (d) clusters of non-ground LIDAR points on buildings and large trees; (e) extracted planar segments from Cluster 1 in (d); and (f) refined segments.



**Figure 3.** Finding coplanar points on a sample scene from the Aitkenvale data set with (a) original point density at 40 points/m<sup>2</sup>; (b) re-sampled point density at 11 points/m<sup>2</sup> and (c) four points/m<sup>2</sup>; and (d) four points/m<sup>2</sup> for the sample scene in Figure 2a. Yellow: coplanar; magenta: non-coplanar; blue: other points.



### 3.1. LIDAR Classification and Mask Generation

If a bare-earth DEM is not available, one can be generated from the LIDAR point cloud data. We assume that the bare-earth DEM is given as an input to the proposed technique. Then, for each LIDAR point, the corresponding DEM height is used as the ground height,  $H_g$ . A height threshold  $T_h = H_g + h_c$ , where  $h_c$  is a height constant that separates low height objects from higher height objects,

is then applied to the LIDAR data. Consequently, the point cloud is divided into two groups. The first group contains the ground points, such as points on the ground, road furniture and bushes that are below  $T_h$ . The second group consists of the non-ground points that represent elevated objects, such as buildings and trees with heights above  $T_h$ . Many authors (e.g., [37,38]) have used  $h_c = 2.5$  m. However, for the Vaihingen data set, it was observed that this height threshold removes many low buildings and sheds. Therefore,  $h_c$  has been set at 1 m for this study. This threshold value will then classify many points, mainly on bushes and low height trees, as the non-ground points. These could otherwise be removed using a high threshold value. Thus, an additional number of unwanted planes may be extracted, and they will be removed using the procedure to be discussed in Section 3.5.

The primary or building mask,  $M_g$ , as shown in Figure 2b, is generated from the ground points following the procedure in [38].  $M_g$  indicates the void areas where there are no laser returns below  $T_h$ , *i.e.*, ground areas covered by buildings and trees. Awrangjeb *et al.* [38] have shown that buildings and trees are found to be thinner in  $M_g$  than in the nDSM (normalized DSM, which indicates the filled areas, from where the laser reflects, above the same threshold, *i.e.*, roofs and tree tops). Consequently,  $M_g$  is used to cluster the non-ground points, as discussed below.

### 3.2. Clustering Non-Ground LIDAR Points

The mask,  $M_g$ , is divided into a  $4 \times 4$ -pixel grid. Since the resolution of  $M_g$  is kept fixed at 0.25 m, independent of the LIDAR point density [38], each of the grid cells occupies an area of  $1 \text{ m}^2$  (same as the minimum roof plane area,  $a_p$ ). Figure 2c shows the grid cells, which can be categorized into three groups: Groups A, B and C. The cells in Group A (see the magenta dots in Figure 2c) contain only the black pixels and represent the areas inside the building or tree boundaries. In contrast, the cells in Group B (the cyan dots in Figure 2c) have both white and black pixels and indicate the areas along the building and tree boundaries. The cells in Group C contain only the white pixels and represent the ground (the empty cells in Figure 2c).

The Group A cells are now separated into clusters, such that any two cells in each cluster are connected by other Group A cells belonging to the same cluster. As shown in Figure 2c, there are five such clusters in the sample scene. Thereafter, the Group B cells along the cluster boundary are added to each of the clusters. Each of the added Group B cells should have at least one Group A cell as its neighbour (a  $3 \times 3$  neighbourhood is considered) in the cluster.

Finally, for each of the clusters, the cluster boundary (boundary points) is obtained using the Canny edge detector. Lines along the boundary are also extracted following the procedure in [38]. First, corners are detected along the extracted boundary. Then, a best-fit straight line is determined via least squares using the boundary points between two successive corners. These lines help to locate roof planes near the building boundary. The non-ground LIDAR points within the cluster boundary are assigned to the cluster. Figure 2d shows the clusters, their boundaries and non-ground LIDAR points for the sample scene.

The clustering technique helps in the elimination of trees that are not dense (e.g., the tree at the bottom of Figure 2) and/or small in area. In addition, dense vegetation can be separated into small parts (e.g., trees at the top-left corner of Figure 2). In a small vegetated area (e.g., Cluster 4 in Figure 2d),

it will be impossible to construct a large plane. Thus, many planes constructed on trees can be easily removed by applying the minimum plane size, as will be described in Section 3.6.

### 3.3. Finding Coplanar Points

Since the final results of a region growing algorithm may be affected by the selection of initial points, there is a division into coplanar and non-coplanar points. The coplanar points are more suitable to be used as seed points than the non-coplanar points. An individual coplanar point constructs a small seed region with its neighbouring points. Such a seed region can then be extended in a region growing fashion to extract a complete planar segment.

By using the Delaunay triangulation algorithm, a natural neighbourhood of non-ground LIDAR points can be generated for either one cluster at a time or all non-ground points at the same time. The neighbourhood of a point,  $P$ , consists of the points,  $Q_i$ ,  $1 \leq i \leq n$ , where each line,  $PQ_i$ , is a side of a Delaunay triangle. In order to avoid the Delaunay neighbours, which are far away from  $P$ , only the neighbours for which  $|PQ_i| \leq 2d_{max}$  are chosen, where  $d_{max}$  is the maximum point spacing in the data.

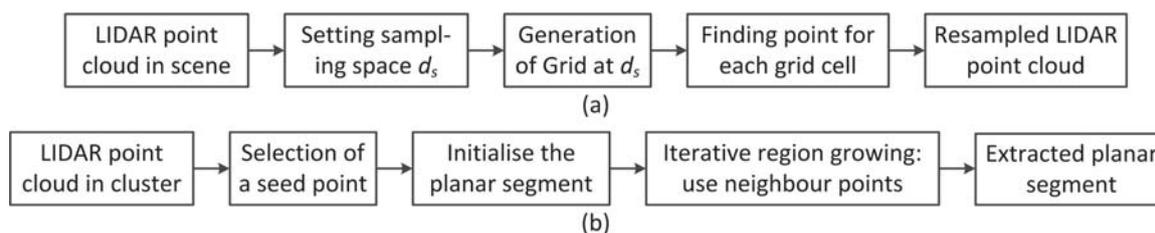
The coplanarity of  $P$  with respect to its neighbouring points is ascertained following the procedure in [15], which is based on eigenvalue analysis. However, in this investigation, it has been observed that while this procedure works well when the point density is low (e.g., four points/m<sup>2</sup>), it is less suitable at a high point density, because the number of non-coplanar points on a nominally planar surface increases. This observation supports the statement in [17] that local surface normals are quite noisy in dense data sets. In cases where the material of a planar roof segment might be tiles or corrugated iron sheets, there will be small difference in heights among neighboring LIDAR points on the same plane. Moreover, there are generally some errors in the LIDAR generated heights. Small errors in neighbouring points of a given point are noticeable (*i.e.*, the plane fit is poor using the procedure in [15]) in dense point clouds where points are close to one another. As shown in Figure 3a, in the sample scene of the Aitkenvale data set, there are many non-coplanar points (magenta dots) on many planes, especially on the right-hand side plane. The number of non-coplanar points on a plane decreases with a decrease in point density. Figure 3b,c illustrates this phenomenon at two different point densities (11 and four points/m<sup>2</sup>, corresponding to the sampled and re-sampled point spacing  $d_s = 0.3$  and  $0.5$  m, respectively).

The coplanarity decision therefore depends on the area of the seed region represented by  $P$  and its neighbours. At a high point density, the seed region is small, and therefore, the procedure may fail. At a low point density, the area of the seed region is large. Thus, the procedure has a high chance of success. Since at a low point density, there might not be the expected number of points on a small plane,  $d_s$  may be set at  $0.5$  m (four points/m<sup>2</sup>) based on the assumption that the smallest plane on a roof is  $a_p = 1$  m<sup>2</sup> in area. Figure 3d shows that this point spacing has a negligible effect on coplanar point determination within the Vaihingen data set, since its original point density is low. The blue dots in Figure 3d are the points that are excluded during the sampling procedure. The sampling procedure, to be discussed below, chooses the points at a given point spacing,  $d_s$  ( $d_s \geq d_{max}$ ). At a high  $d_s$  value, the number of chosen points will be small. Thus, the point density of the input LIDAR point cloud is decreased.

The flow diagram in Figure 4a shows the resampling procedure. First, the sampling space,  $d_s$ , is set as follows. If the input LIDAR density is at least four points/m<sup>2</sup> (*i.e.*,  $d_{max} \leq 0.5$  m), a sampling procedure

is followed at  $d_s = 0.5$  m. If the point density is less than four points/m<sup>2</sup>, the sampling procedure is followed using  $d_s = d_{max}$ . In order to find the sampled LIDAR points, a grid is then generated at a corresponding resolution of  $d_s$ . Finally, all the LIDAR points within a square grid cell are found, and the point which is closest to the centre of the cell is chosen as the sampled point for the cell. Cells which do not contain any points are left empty. The coplanarity of all the sampled points is determined by setting  $d_{max} = d_s$ .

**Figure 4.** Flow diagrams for: (a) resampling of the LIDAR point cloud of a scene; and (b) extraction of a planar segment from the LIDAR point cloud of a cluster.



### 3.4. Extracting Planar Segments

The planar segments are iteratively extracted from the LIDAR point cloud within each of the clusters obtained through the procedure described in Section 3.2. Figure 4b shows an iteration, where a planar segment is initialised and extended in a region growing fashion, which will now be described.

Let the two sets of the non-ground LIDAR points from each cluster be  $S_1$ , containing all the coplanar points, and  $S_2$ , containing the rest (non-coplanar and excluded points during sampling). An unused coplanar point,  $P \in S_1$ , is first selected as a seed point, and then a planar segment is initialised using  $P$  and the neighbours of  $P$ . Let  $S_p$  be the set of points consisting of  $P$  and its neighbours. An initial plane,  $\xi$  (equation:  $\alpha x + \beta y + \gamma z = \delta$ ) is estimated using points in  $S_p$ , so long as  $|S_p| \geq 3$ .

The random selection of  $P$  can result in a number of unexpected planes. In order to limit the number of such planes,  $P$  can be initially located along the cluster boundary using the boundary lines extracted via the method described in Section 3.2. For example,  $P$  can be the nearest coplanar point from the midpoint of a boundary line. Later, when no coplanar points are found along the boundary lines, an unused coplanar point is randomly selected, and a new planar segment is grown.

The new planar segment is now iteratively extended using the neighbouring points from  $S_1$  and  $S_2$ . A neighbour point set  $S_n \subseteq \{S_1 \cup S_2\}$  is found for  $S_p$  within a distance threshold of  $T_d = 2d_{max}$ . A plane compatibility test is executed for each point,  $Q \in S_n$ , using either of the following two approaches: the estimated height of  $Q$  using  $\xi$  is similar to its LIDAR height (within the flat threshold  $T_f = 0.10$  m introduced in [7]); or, the normal distance of  $Q$  to the plane is at most  $T_p = 0.15$  m [7]. If  $Q$  passes the plane compatibility test, it is included into  $S_p$ , and  $\xi$  is updated; otherwise, it is discarded. Once all points in  $S_n$  are determined, a new  $S_n$  is found, and the plane is iteratively extended, until no further points can be added to  $S_p$ .

Once the segment extension is complete, all the coplanar points within it are marked, so that none will later be used for initiating another planar segment. As a result, the points in  $S_2$ , which mainly reside along the plane boundaries, can be used by more than one plane. The next planar segment is grown

by using an unused coplanar point (from  $S_1$ ) and following the same procedure discussed above. The iterative procedure continues, until no coplanar point remains unused.

Figure 2e shows all the extracted planes from Cluster 1 of Figure 2d. There were 14 extracted planes. Many of the extracted planes overlap the neighbouring planes. Most importantly, one of the planes was wrongly extracted, as shown by red dots in Figure 2e. The wrong plane was initialised by a coplanar point near to the ridge line, where two planes intersect. As shown in the magnified part in Figure 2e, the neighbours of the coplanar point reside on the two neighbouring planes. Consequently, the wrongly extracted plane includes points from a total of six planes.

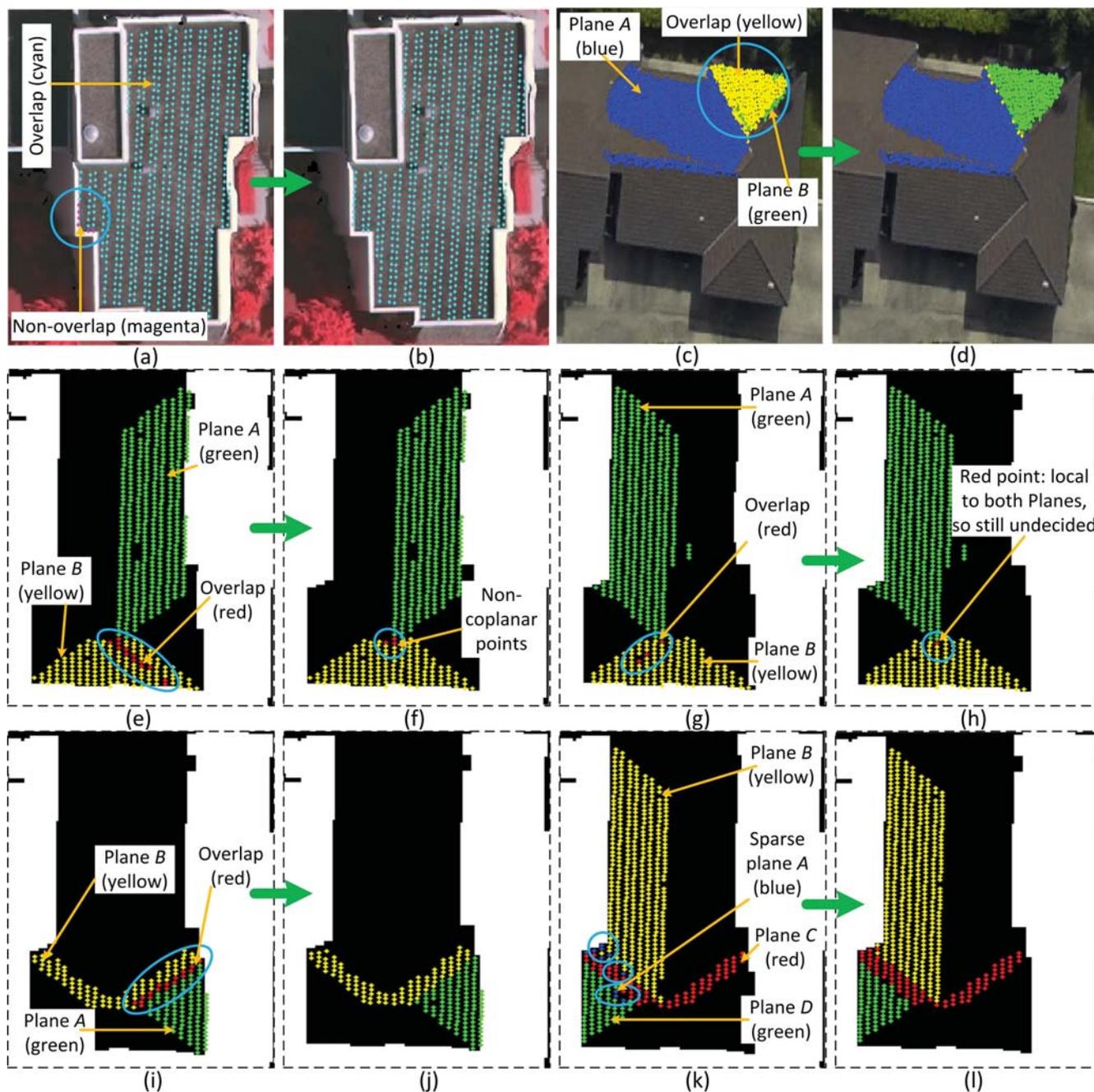
### 3.5. Refining Planar Segments

If two extracted planes have one or more LIDAR points in common, they may be coincident, parallel or non-parallel planes. Moreover, a wrongly extracted plane, shown in Figure 2e, may intersect one or more additional extracted plane and, thus, may have points in common with the intersected planes. In order to refine the extracted planes, the common or overlapping points must be assigned to the most appropriate planes. However, since the extracted planes are estimated from the input LIDAR points, which usually have some accuracy limitation, it is hard to find the exact relationship (e.g., parallelism, intersection lines and angle) between two planes that have overlapping points. In this study, the topology or relationship between two such planes is obtained by using the overlapping points and the intersection line between the planes.

The following six steps are executed sequentially for any two planes that overlap each other by at least one point. Let the two extracted planes,  $A$  and  $B$ , consist of the LIDAR point sets,  $S_a$  and  $S_b$ , respectively. Let  $S_o$  be the set of overlapping points between them. At any instance of the refinement procedure, the plane equations ( $\xi_a$  and  $\xi_b$ ) and normals ( $\eta_a$  and  $\eta_b$ ) are estimated using the non-overlapping points from  $A$  ( $S_c = S_a/S_o$ ) and  $B$  ( $S_d = S_b/S_o$ ). Let the normal distances from a point  $P_o \in S_o$  to the planes be  $\ell_{oa}$  and  $\ell_{ob}$ . Further, let the number of points that are close (within  $T_d$ ) to  $P_o$  from  $S_c$  and  $S_d$  be  $n_a$  and  $n_b$ , respectively. The locality of  $P_o$  with respect to  $A$  is defined by  $n_a$ . If  $P_o$  is far away (more than  $T_d$ ) from  $S_c$ , then  $n_a = 0$ . However if  $P_o$  has neighbouring points only from  $S_c$ , then  $n_a$  is high.

- Step 1: Merging (coincident) planes. A plane may be extracted more than once. In this case, the two planes share almost all the points on the plane. If  $A$  and  $B$  overlap each other by at least 90%, they are merged into a single plane. Figure 5a shows an example (from the Vaihingen data set) where the cyan coloured points are common to two extracted planes, and the magenta coloured points within the light blue circle are the only difference. Figure 5b shows the merged plane.
- Step 2: Parallel planes. If  $A$  and  $B$  are parallel (when the angle,  $\theta$ , between  $\eta_a$  and  $\eta_b$  is at most  $T_\theta = \frac{\pi}{32}$ ), each point,  $P_o \in S_o$ , is assigned based on its normal distances to the planes and locality. If  $\ell_{oa} < \ell_{ob}$  and  $n_a > n_b$ ,  $P_o$  is assigned to Plane  $A$ . If  $\ell_{oa} > \ell_{ob}$  and  $n_a < n_b$ ,  $P_o$  is assigned to Plane  $B$ . Otherwise,  $P_o$  still remains unassigned and will be assigned in Steps 4 or 5 below. As shown in Figure 5c,d, the overlapping region between the two parallel planes,  $A$  and  $B$  ( $\theta = 3^\circ$ ), is rectified accordingly. The unassigned points, shown in yellow in Figure 5d, will be dealt with in Steps 4 or 5 below.

**Figure 5.** Refinement of planar segments: (a,b) coincident planes; (c,d) parallel planes; (e,f) non-parallel planes; (g,h) using locality; (i,j) using plane intersection line; and (k,l) the split and merge of a sparse plane.



- Step 3: Non-parallel planes. If  $A$  and  $B$  are not parallel (*i.e.*,  $\theta > \frac{\pi}{32}$ ), each coplanar point,  $P_o \in S_o$ , is determined as follows. Let the normal at  $P_o$  be  $\eta_o$  and the angles between  $\eta_o$  and the two plane normals ( $\eta_a$  and  $\eta_b$ ) be  $\alpha_{oa}$  and  $\alpha_{ob}$ , respectively. If  $\alpha_{oa} < \alpha_{ob}$ ,  $P_o$  and its neighbours are assigned to Plane  $A$ ; otherwise, to Plane  $B$ . The coplanar points among the overlapping points in Figure 5e are assigned to Plane  $B$ , but the non-coplanar points remain unassigned, as shown in Figure 5f. The unassigned points will be dealt with in Steps 4 or 5 below.
- Step 4: Using locality. If  $P_o$  resides within Plane  $A$ , but away from Plane  $B$ , *i.e.*,  $n_a > 0$  and  $n_b = 0$ , then  $P_o$  is assigned to Plane  $A$ . Conversely, if  $P_o$  resides within Plane  $B$ , but away from

Plane  $A$ , *i.e.*,  $n_b > 0$  and  $n_a = 0$ , then  $P_o$  is assigned to Plane  $B$ . Figure 5g,h shows that two overlapping points have been assigned to Plane  $B$  based on locality, but one point still remains unassigned, as this is local to both of the planes.

- Step 5: Using plane intersection. All the unassigned overlapping points are finally assigned using the plane intersection line (between Planes  $A$  and  $B$  using  $\xi_a$  and  $\xi_b$ ). If  $P_o$  and Plane  $A$  reside on the same side of the intersection line, then  $P_o$  is assigned to Plane  $A$ ; otherwise, to Plane  $B$ . However, if  $A$  and  $B$  are parallel, the intersection line is not used, and  $P_o$  is assigned based on its normal distances ( $\ell_{oa}$  and  $\ell_{ob}$ ) to the planes. If  $\ell_{oa} < \ell_{ob}$ ,  $P_o$  is assigned to Plane  $A$ ; otherwise, to Plane  $B$ . Figure 5i,j shows an example where the overlapping points between two non-parallel planes are assigned based on the intersection line.
- Step 6: Split and merge. After the execution of all the above steps, an extracted plane,  $A$ , may be sparse in the sense that it may have some points that reside away from other points in the same plane. Each sparse plane is split into two or more parts ( $A_1, A_2, A_3, \dots$ ), where points in each part reside at a distance of at least  $T_d$  from points in the other parts. Figure 5k shows such a sparse plane in blue, which is split into three parts. For each point,  $P_o$ , in each part, the nearest extracted plane,  $B$ , is obtained based on its normal distances to all the neighbouring planes (within a distance of  $T_d$ ). If the normal distance to  $B$  (*i.e.*,  $\ell_{ob}$ ) is smaller than  $T_p = 0.15$  m, then  $P_o$  is assigned to Plane  $B$ . If such a plane cannot be found (*i.e.*,  $\ell_{ob} > T_p$ ) for some or all the points forming a part, then this part remains as a separate extracted plane. Figure 5l shows that all the points of the sparse plane are well assigned to the respective neighbouring planes.

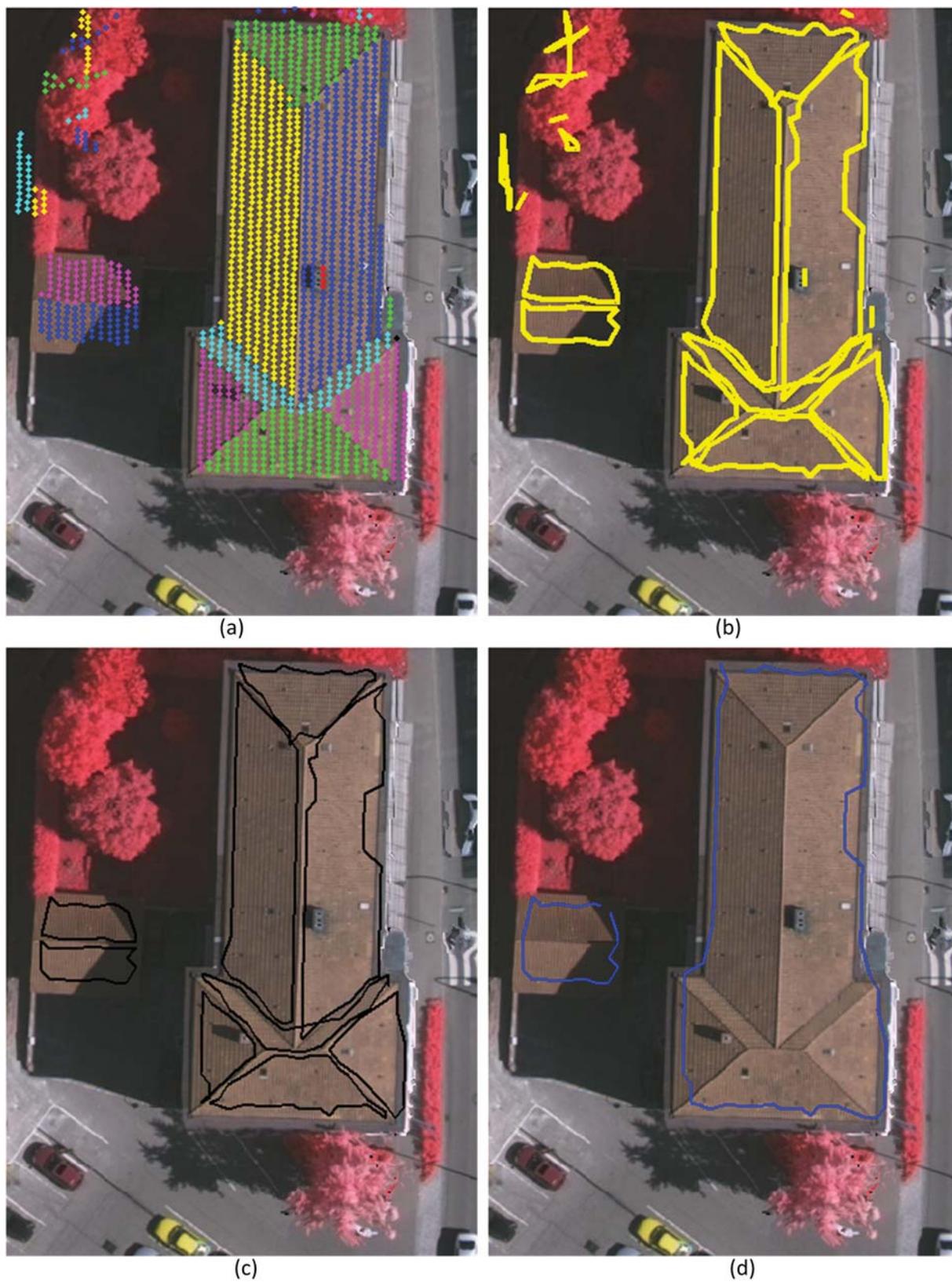
Figure 2f shows the extracted planar segments for Cluster 1 in Figure 2d following the above refinement procedure.

### 3.6. Removing False Planes

Figure 6a,b shows all the extracted planar segments and their boundaries on the sample scene shown in Figure 2a. In order to find the boundary of a plane, the corresponding set of LIDAR points,  $S_p$ , is used to generate a binary mask,  $M_b$  (similar to the building mask,  $M_g$ ). The boundary of the plane is the Canny edge around the black shape in  $M_b$ . For each edge point, the nearest LIDAR point height from  $S_p$  is assigned.

In order to remove false positive planes, mostly constructed on trees, a new rule-based procedure is proposed. For each extracted plane, the procedure uses information, such as plane area, straight line segments along the plane boundary, the relation with the neighbouring planes, random point spikes, unused (not on any estimated plane) LIDAR points and the average height difference among the LIDAR points within the boundary. Moreover, the number of used LIDAR points within the corresponding cluster of the plane is employed. The routines for estimating area, straight lines, random point spikes and neighbouring planes have been adopted from [7]. In order to obtain the average height difference within an extracted plane, the mean height difference for each non-ground LIDAR point is first calculated with its neighbouring points (the sampled LIDAR data in Section 3.3 is used). The average height difference for the plane is then estimated for all LIDAR points within the plane boundary, including those that are not on the plane.

**Figure 6.** Removing false planes: (a) all extracted planes; (b) all plane boundaries; (c) final plane boundaries; and (d) building boundaries.



Initially, all the extracted planes are marked as true roof planes. Then, in order to remove the false positive planes, the following tests are executed sequentially. All the parameter values are listed in Table 1.

**Table 1.** Parameters used by the proposed roof extraction method (“this paper” indicates that a corresponding parameter value has been chosen in this study).

Parameters	Values	Sources
Ground height $H_g$	DEM height	input LIDAR data
Height threshold $T_h$	$H_g + 1.0$ m	this paper
Mask resolution	0.25 m	[38]
Straight line length	3 m	[38]
Maximum point spacing $d_{max}$	(from input LIDAR data)	input LIDAR data
Point neighbourhood $T_d$	$2d_{max}$	related to $d_{max}$
Flat height threshold $T_f$	0.10 m	[7]
Normal distance threshold $T_p$	0.15 m	[7]
Shared points for coincident planes	$\geq 90\%$	this paper
Angle between parallel planes $T_\theta$	$\frac{\pi}{32}$	this paper
Minimum roof plane area $a_p$	1 m <sup>2</sup>	this paper
Minimum roof plane width	1 m	this paper
Small plane area threshold $T_{a1}$	2 m <sup>2</sup>	this paper
Medium plane area threshold $T_{a2}$	5 m <sup>2</sup>	this paper
Unused point ratio $r$	10%, 35%	this paper
Used point ratio $r_u$	60%	this paper
Average height differences $T_{hd}$	0.8, 0.5, 0.2 m	this paper
Spike height threshold $T_z$	1.5 m	[7]
Number of random points for spike test	10	[7]

- Area test: If the area of an extracted plane,  $A$ , is less than  $a_p = 1$  m<sup>2</sup>, it is marked as a false plane.
- Random spike test: A 3D cuboid is considered around  $A$  using its minimum and maximum Easting, Northing and height values. Let the minimum and maximum heights of the cuboid be  $z_m$  and  $z_M$ . Then, a number (10, in this study) of random 2D points are generated, which are uniformly distributed within the cuboid, and their height values are estimated using the plane equation,  $\xi_a$ . Usually, a small number of random points are enough for this test. A large number of points will slow down the algorithm, so the selection of 10 random points per plane has been accepted in this study [7]. If at least one estimated height,  $z_e$ , is too low ( $z_m - z_e > T_z$ ) or too high ( $z_e - z_M > T_z$ ) and the area of  $A$  is less than an area threshold,  $T_{a1}$ , then  $A$  is classified as a false plane.
- Unused LIDAR point test: The ratio,  $r$ , of the number of unused LIDAR points (not found on any extracted planes) to the number of used LIDAR points ( $S_p$ ) within  $A$ 's boundary is calculated. If  $r \geq 10\%$  for a small plane (smaller than  $T_{a2}$ ) or if  $r \geq 35\%$  for a medium plane (smaller than  $3T_{a2}$ ), then  $A$  may be marked as a false plane. However, if  $A$  is at least 1 m in width, has at least one long straight line segment of at least 3 m in length and its average height difference is less than  $T_{hd1}$ , then  $A$  is not marked as false.

- Average height test: If the area of  $A$  is less than  $T_{a2}$  and its average height difference is more than  $T_{hd1}$ , then  $A$  is a tree plane.
- Used LIDAR point test: The ratio,  $r_u$ , of the number of used LIDAR points to the total number of non-ground LIDAR points within a cluster boundary is calculated. If  $r_u < 60\%$ , then all the unmarked planes belonging to this cluster are further tested. If  $A$  does not have a straight line segment of at least 3 m in length along its boundary and its width is less than 1 m, then  $A$  is marked as a tree plane.

All planes are subjected to the tests above. If  $A$  is a plane that was marked to be removed as false, then for an unmarked plane,  $B$ , which is a neighbour of  $A$ , the following first three tests are executed sequentially:

- With neighbours, Test 1: If  $B$  has unused points inside its boundary ( $r \geq 10\%$ ) and its average height difference is greater than  $T_{hd2}$ , then it is marked as false, as well. This test continues until none of the remaining planes can be marked as false.
- With neighbours, Test 2: If  $B$  is less than  $2T_{a2}$  in area and all of its neighbours are already marked as false, then it may also be a false plane. If  $B$  has parallel or perpendicular line segments along its boundary and its average height difference is small (less than  $T_{hd3}$ ), it is determined to be a true plane. Otherwise, it is marked as a false plane. This test continues, until none of the remaining planes can be marked as false.
- Planes reside inside one another, Test 3: If  $A$  has been classified as false in any of the above tests and if it resides within the 2D plane boundary of a bigger plane, which has not been determined to be false, then  $A$  is marked as true. This test helps the retention of dormers as true roof planes, though they are small in area.

Figure 6c shows all the roof planes obtained after this removal procedure for the sample test scene. An individual building can now be easily obtained as a group of planes. All the LIDAR points from a group of planes are used together to extract the corresponding building boundary. Figure 6d shows the extracted building boundaries.

#### 4. Performance Study

In the performance study conducted to assess the proposed approach, six data sets from different geographic locations were employed. The objective evaluation followed both the threshold-based system [39] adopted by the ISPRS (International Society for Photogrammetry and Remote Sensing) benchmark project [40] and an automatic and threshold-free evaluation system [41].

In both evaluation systems [39,41], three categories of evaluations (object-based, pixel-based and geometric) have been considered. A number of metrics are used in the evaluation of each category. While the object-based metrics (completeness, correctness, quality, under- and over-segmentation errors and detection and reference cross-lap rates) estimate the performance by counting the number of objects (buildings or planes in this study), the pixel-based metrics (completeness, correctness, quality, area omission and commission errors) show the accuracy of the extracted objects by counting the number of pixels. In addition, the geometric metric (root mean square error, RMSE) indicates the accuracy of

the extracted boundaries and planes with respect to the reference entities. The definitions and how these metrics are estimated have been adopted from [16,39,41]. Once the reference and extracted building and plane boundaries are obtained, these metrics are automatically determined via the performance evaluation techniques [39,41]. In the ISPRS benchmark [40], the minimum areas for large buildings and planes have been set at 50 m<sup>2</sup> and 10 m<sup>2</sup>, respectively. Thus, the object-based completeness, correctness and quality values will be separately shown for large buildings and planes.

In addition, the efficiency of the proposed method will be shown in terms of data processing time. Since the involved test areas differ in size and point density, computation time per area and computation time per point will also be estimated.

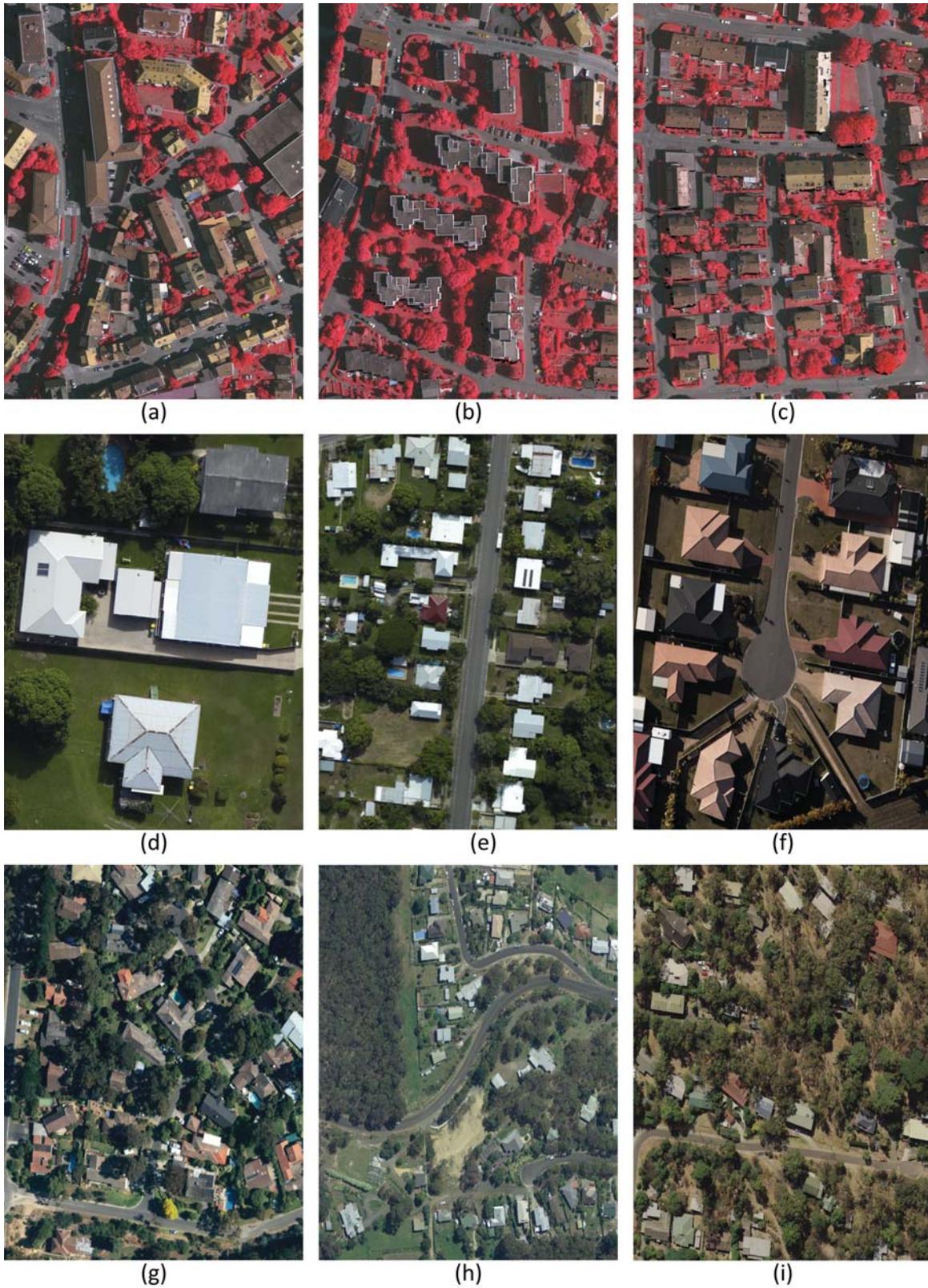
#### 4.1. Data Sets

The first data set is Vaihingen (VH) [36], from the ISPRS benchmark [40]. There are three test sites in this data set (Figure 7a–c) and each area is covered with a point density of four points/m<sup>2</sup>. Area 1 is characterised by dense development consisting of historic buildings having complex shapes. Area 2 is characterised by a few high rise residential buildings surrounded by trees. Area 3 is purely residential, with detached houses and many surrounding trees. The number of buildings (larger than 2.5 m<sup>2</sup>) in these three areas is 37, 14 and 56, and the corresponding number of planes is 288, 69 and 235, respectively.

The second data set is Aitkenvale (AV), which has a high point density (29 to 40 points/m<sup>2</sup>), and the third data set is Hervey Bay (HB), which has a medium point density (12 points/m<sup>2</sup>) [7]. There were two sites in the AV data set, as shown in Figure 7d,e. The first (AV 1) covers an area of 66 m × 52 m, has a point density of 40 points/m<sup>2</sup> and contains five buildings comprising 26 roof planes. The second AV site (AV 2) has a point density of 29 points/m<sup>2</sup>, covers an area of 214 m × 159 m and contains 63 buildings (four were between four to 5 m<sup>2</sup> and 10 were between five to 10 m<sup>2</sup>), comprising 211 roof planes. The single site of the HB data set in Figure 7f covers 108 m × 104 m and contains 28 buildings (three were between four to 5 m<sup>2</sup> and six were between five to 10 m<sup>2</sup>), consisting of 166 roof planes. These three sites contain mostly residential buildings, and they can be characterized as urban with medium housing density and moderate tree coverage that partially covers buildings. In terms of topography, AV is flat, while HB is moderately hilly. For all three data sets, bare-earth DEMs of 1-m horizontal resolution were available.

The other three data sets, Eltham (EL), Hobart (HT) and Knox (KN), have point densities of five, two and one points/m<sup>2</sup>, respectively. The three test areas, shown in Figure 7g–i, cover 393 m × 224 m, 303 m × 302 m and 205 m × 204 m, respectively. The EL data set contains 75 buildings (nine were less than 10 m<sup>2</sup>, including five within three to 5 m<sup>2</sup>) consisting of 441 planes (46 were less than 5 m<sup>2</sup>). The HT data set has 69 buildings (13 were less than 10 m<sup>2</sup>, including four within one to 5 m<sup>2</sup>) containing 257 planes (24 less than 5 m<sup>2</sup>). The KN data set contains 52 buildings (eight were less than 10 m<sup>2</sup>, including four within two to 5 m<sup>2</sup>) consisting of 181 planes (48 were less than 10 m<sup>2</sup>, including 24 less than 5 m<sup>2</sup>). These three data sets have dense vegetation and are in hilly areas. Many of the buildings are severely occluded by the surrounding trees. Moreover, in the KN data set, some parts of the building are at a similar height to the surrounding sloping grounds. Consequently, these parts could not be found in the building mask.

**Figure 7.** Data sets: Vaihingen (a) Area 1; (b) Area 2 and (c) Area 3; Aitkenvale (d) Area 1 and (e) Area 2; (f) Hervey Bay; (g) Eltham (partially shown, full area is in Figure 8a; (h) Hobart; and (i) Knox.



The results on the VH data set have been evaluated using the threshold-based evaluation system [39]. Evaluation results for the VH data set have been confirmed by the ISPRS Commission III, Working Group 4, and are available at [42,43] (see the method labeled as “MON”). For all other data sets, 2D reference data were created by monoscopic image measurement using the Barista software [44]. Evaluation results and data sets are available at [45]. All visible roof planes were digitized as polygons irrespective of their size. The reference data included garden sheds and garages. These were sometimes as small as 1 m<sup>2</sup> in area.

#### 4.2. Parameter Setting

Table 1 shows the list of parameters used by the proposed algorithm. Many of these parameter values are either adopted from the existing literature or dependent upon the input LIDAR data. The setting of values for the remaining parameters requires further explanation. For these parameters, standard values have been chosen following a trial and error approach, applied on the first two areas of the Vaihingen data set. The application of the adopted values to seven additional areas from six different data sets having dissimilar point densities subsequently indicated that the parameter values are generally insensitive to LIDAR point density. Note that only the parameter,  $d_{max}$ , which is dependent upon the input LIDAR point cloud, needs to be changed across different data sets. The point neighbourhood,  $T_d$ , will be automatically altered once  $d_{max}$  is changed. All other parameters are kept unchanged across the different data sets.

The value of the height threshold,  $T_h$ , was  $H_g + 2.5$  m, which Awrangjeb *et al.* [7] had previously employed for the AV and HB data sets. Since there may be some buildings (especially in the VH data set) with low roof heights,  $T_h = H_g + 1$  m has been set in this study, and it also works well for the AV and HB data sets.

For the coincident planes (when a roof plane is extracted twice), it is assumed that they share the majority (at least 90%) of plane points, and the remaining points may be included from small roof structures, such as chimneys or roof overhangs. The angle tolerance between two parallel planes,  $\theta$ , is set at  $\frac{\pi}{32}$ , which is much smaller than the  $\frac{\pi}{8}$  tolerance, previously used for obtaining the parallel lines [38].

Planes extracted on trees are usually small in size, and so, large roof planes are easily distinguishable. Moreover, planes smaller than 1 m<sup>2</sup> in size may not have enough LIDAR points for segmentation. Therefore, setting the minimum roof plane area,  $a_p$ , and width at 1 m<sup>2</sup> and 1 m, respectively, is quite reasonable. Firstly, planes smaller than 1 m<sup>2</sup> may not have enough LIDAR points (*i.e.*, at least three points are required to initiate the plane equation, and at least one of the points should be coplanar with its neighbours to form a seed region). Moreover, in this study, if an extracted plane smaller than 1 m<sup>2</sup> is a neighbour of a large plane, then the small plane is kept as a true roof plane. This helps to extract chimneys and other small planes on a building roof. The size of the small roof planes is set at between one and 5 m<sup>2</sup> (*i.e.*,  $>a_p$  and  $\leq T_{a2}$ ) and that of medium planes is between five and 15 m<sup>2</sup> (*i.e.*,  $>T_{a2}$  and  $\leq 3T_{a2}$ ). Any random point spike within an extracted small plane, which is  $\leq T_{a1} = 2$  m<sup>2</sup> in area, indicates that this plane may be a tree plane. The number of the unused points and the height difference among the points within a medium tree plane can be high. As the area of a tree plane increases, the number of unused points on the tree plane also increases. Therefore, the ratio,  $r$ , of unused to used points is usually low

for small tree planes and high for medium tree planes. In this study,  $r$  is set at 10% for small planes and at 35% for medium planes. Conversely, the ratio,  $r_u$ , of used points (for plane extraction) to the total number of points on a non-ground object is higher for a building roof than for a tree. The value of  $r_u$  is set at 60% irrespective of the object size.

The average height difference,  $T_{hd}$ , among points on a tree plane is generally greater than that on a roof plane.  $T_{hd}$  can be small for a small area of a tree, but it increases for a large tree area. Therefore, if  $T_{hd}$  is more than 0.8 m on a plane, then the plane is removed irrespective of its size. However, for small tree planes, which have many unused points, as well ( $r > 10\%$ ),  $T_{hd}$  is set at 0.2 m. If a roof plane is partially occluded by the nearby trees, then  $T_{hd}$  may also be large for that plane. In order to keep such a plane, the straight line segments along the plane boundary are obtained. If the lines are parallel or perpendicular, then the plane is kept, even if  $T_{hd}$  grows to 0.5 m.

### 4.3. Results and Discussions

The results and discussions on the test data sets are presented separately for building detection and roof plane extraction.

#### 4.3.1. Building Detection Results

Tables 2 and 3 show the object- and pixel-based evaluation results, respectively, for the VH data set using the evaluation system of [39]. Tables 4 and 5 show the same for the other data sets, which were evaluated using the evaluation system of [41]. Figure 8 shows the detection results on the EL data set. Figure 9 shows the same for Area 3 of the VH and Area 2 of the AV data sets. Both figures also show some detection examples from the three test areas. Since the evaluation system in [39] does not provide the area omission and commission errors, these errors, shown in Table 3, for the three areas of the VH data sets were evaluated following the procedure in [41].

The proposed algorithm missed some small buildings in all three areas of the VH data set. However, the large buildings were correctly extracted, as is evident from the completeness, correctness and quality indices for buildings over 50 m<sup>2</sup> in Table 2. A large building in the top-middle of Area 3 (shown within the black circle in Figure 9a) was only partially detected, due to information missing in the LIDAR data, and it was classified as a false negative by the evaluation system. That is why the completeness value for buildings larger than 50 m<sup>2</sup> in Table 2 is not 100% for Area 3. The under-segmentation cases occurred when nearby buildings were found merged together. As shown in Figure 9g, two car ports were merged with neighbouring buildings. This unexpected merging could be avoided by analysing white pixels in between the black shapes in the building mask (Section 3.1). In all three areas, the omission errors were higher than the commission errors. This performance indicates that while the proposed method missed some of the small buildings and parts of the larger buildings, it successfully removed most of the vegetation. The planimetric accuracies were close to one to two times the point spacing of the input LIDAR data.

A similar detection trend was observed in the AV and HB data sets, as shown in Tables 4 and 5. Although there were no under- and over-segmentation cases, as the buildings were well separated from each other, there were omission and commission errors, specially in Area 2 of the AV data set. In this

area, there were many small buildings surrounded by trees. As shown in Figure 9, the proposed algorithm was capable of detecting small buildings, which were partially occluded, but it missed others, which were either significantly occluded (Figure 9d) or had only small planes on their roofs (Figure 9f).

**Table 2.** Building detection results: object-based evaluation for the Vaihingen (VH) data set.  $C_m$  = completeness,  $C_r$  = correctness and  $Q_l$  = quality ( $C_{m,50}$ ,  $C_{r,50}$ ,  $Q_{l,50}$  are for buildings over 50 m<sup>2</sup>) in percentage; 1: $M$  = over-segmentation and  $N$ :1 = under-segmentation,  $N$ : $M$  = both over- and under-segmentation in the number of buildings.

Areas	$C_m$	$C_r$	$Q_l$	$C_{m,50}$	$C_{r,50}$	$Q_{l,50}$	1: $M$	$N$ :1	$N$ : $M$
VH 1	83.8	96.9	81.6	100	100	100	0	6	0
VH 2	85.7	84.6	74.2	100	100	100	0	2	0
VH 3	78.6	97.8	77.2	97.4	100	97.4	0	7	0
<b>Average</b>	<b>82.7</b>	<b>93.1</b>	<b>77.7</b>	<b>99.1</b>	<b>100</b>	<b>99.1</b>	<b>0</b>	<b>5</b>	<b>0</b>

**Table 3.** Building detection results: pixel-based and geometric evaluation for the Vaihingen (VH) data set.  $C_{mp}$  = completeness,  $C_{rp}$  = correctness,  $Q_{lp}$  = quality,  $A_{oe}$  = area omission error and  $A_{ce}$  = area commission error in percentage;  $RMSE$  = planimetric accuracy in metres.

Areas	$C_{mp}$	$C_{rp}$	$Q_{lp}$	$A_{oe}$	$A_{ce}$	$RMSE$
VH 1	92.7	88.7	82.9	13.9	4.6	1.11
VH 2	91.5	91	83.9	15	0.9	0.83
VH 3	93.9	86.3	81.7	15.2	1.8	0.89
<b>Average</b>	<b>92.7</b>	<b>88.7</b>	<b>82.8</b>	<b>14.7</b>	<b>2.4</b>	<b>0.94</b>

**Table 4.** Building detection results: object-based evaluation for the Aitkenvale (AV), Hervey Bay (HB), Eltham (EL), Hobart (HT) and Knox (KN) data sets in percentage.  $C_m$  = completeness,  $C_r$  = correctness and  $Q_l$  = quality ( $C_{m,10}$ ,  $C_{r,10}$ ,  $Q_{l,10}$  are for buildings over 10 m<sup>2</sup>);  $C_{rd}$  = detection cross-lap (under-segmentation) and  $C_{rr}$  = reference cross-lap (over-segmentation) rates.

Areas	$C_m$	$C_r$	$Q_l$	$C_{m,10}$	$C_{r,10}$	$Q_{l,10}$	$C_{rd}$	$C_{rr}$
AV 1	100	100	100	100	100	100	0	0
AV 2	67.2	100	67.2	81.1	100	81.1	0	0
HB	100	100	100	100	100	100	0	0
EL	77.6	88.2	70.3	91.8	88.2	81.8	7.1	1.4
HT	71.2	80.8	60.9	97.7	80.8	79.3	10.8	0
KN	69.2	75	56.3	87.5	68.3	62.2	15	1.9
<b>Average</b>	<b>80.9</b>	<b>90.7</b>	<b>75.8</b>	<b>93</b>	<b>89.6</b>	<b>84.1</b>	<b>5.4</b>	<b>0.6</b>

**Table 5.** Building detection results: pixel-based and geometric evaluation for the Aitkenvale (AV), Hervey Bay (HB), Eltham (EL), Hobart (HT) and Knox (KN) data sets.  $C_{mp}$  = completeness,  $C_{rp}$  = correctness,  $Q_{lp}$  = quality,  $A_{oe}$  = area omission error and  $A_{ce}$  = area commission error in percentage;  $RMSE$  = planimetric accuracy in metres.

Areas	$C_{mp}$	$C_{rp}$	$Q_{lp}$	$A_{oe}$	$A_{ce}$	$RMSE$
AV 1	96	97.4	93.6	4.1	2.6	0.44
AV 2	87.2	94.9	83.2	12.9	5.2	0.66
HB	93	94.5	88.3	7	5.5	0.68
EL	85.6	90.1	78.2	14.4	10	1.31
HT	80	80.2	66.8	20	19.9	1.33
KN	66.2	58	44.8	33.8	42	1.80
<b>Average</b>	<b>84.7</b>	<b>85.9</b>	<b>75.8</b>	<b>15.4</b>	<b>14.2</b>	<b>1.04</b>

**Figure 8.** Building detection on (a) the Eltham data set (five points/m<sup>2</sup>); (b–d) some detection examples on complex cases; and (e–g) some missing buildings in difficult cases. Areas marked by letters in (a) are magnified in (b) to (g).



**Figure 9.** Building detection on (a) Area 3 of the VH data set and (b) Area 2 of the AV data set. Building detection examples in: (c–f) Area 2 of the AV data set; and (g,h) Area 3 of the VH data set. Areas marked by letters in (a) and (b) are magnified in (c) to (h).



Compared to the VH data set, the point density of the AV and HB data sets is high. As shown in Table 2, in the VH data set, the proposed method extracted almost all the buildings larger than 50 m<sup>2</sup>. Table 4 shows that in the AV and HB data sets, the method also extracted all small buildings (*i.e.*, garages and garden sheds), which were as small as 10 m<sup>2</sup>, except in the AV 2 area, where some severely occluded garden sheds (e.g., Figure 9d) were missed. Thus, the completeness for buildings larger than 10 m<sup>2</sup> in Table 4 is not a maximum for the AV 2 area. Nevertheless, the proposed method was able to extract all buildings larger than 50 m<sup>2</sup> in this area.

Compared to the HT data set, the EL and KN areas are more hilly and have more occluded buildings. However, the proposed algorithm performed better in the EL data set than in the HT and KN data sets. This is because the point density in the EL data set is higher than that in the HT and KN data sets. In all three, some large trees could not be removed, as indicated within magenta coloured ellipses in Figures 8 and 10. As a result, the correctness values are less than 90%, even for buildings larger than 10 m<sup>2</sup> in area. Some of the detected trees in the EL and KN data sets were very dense, so laser points hardly penetrated the canopy. Some of the detected tree tops in the HT data set were not only very dense, but also shaped such that they appeared to be flat planes. The extracted planes on these trees were thus too large to be removed.

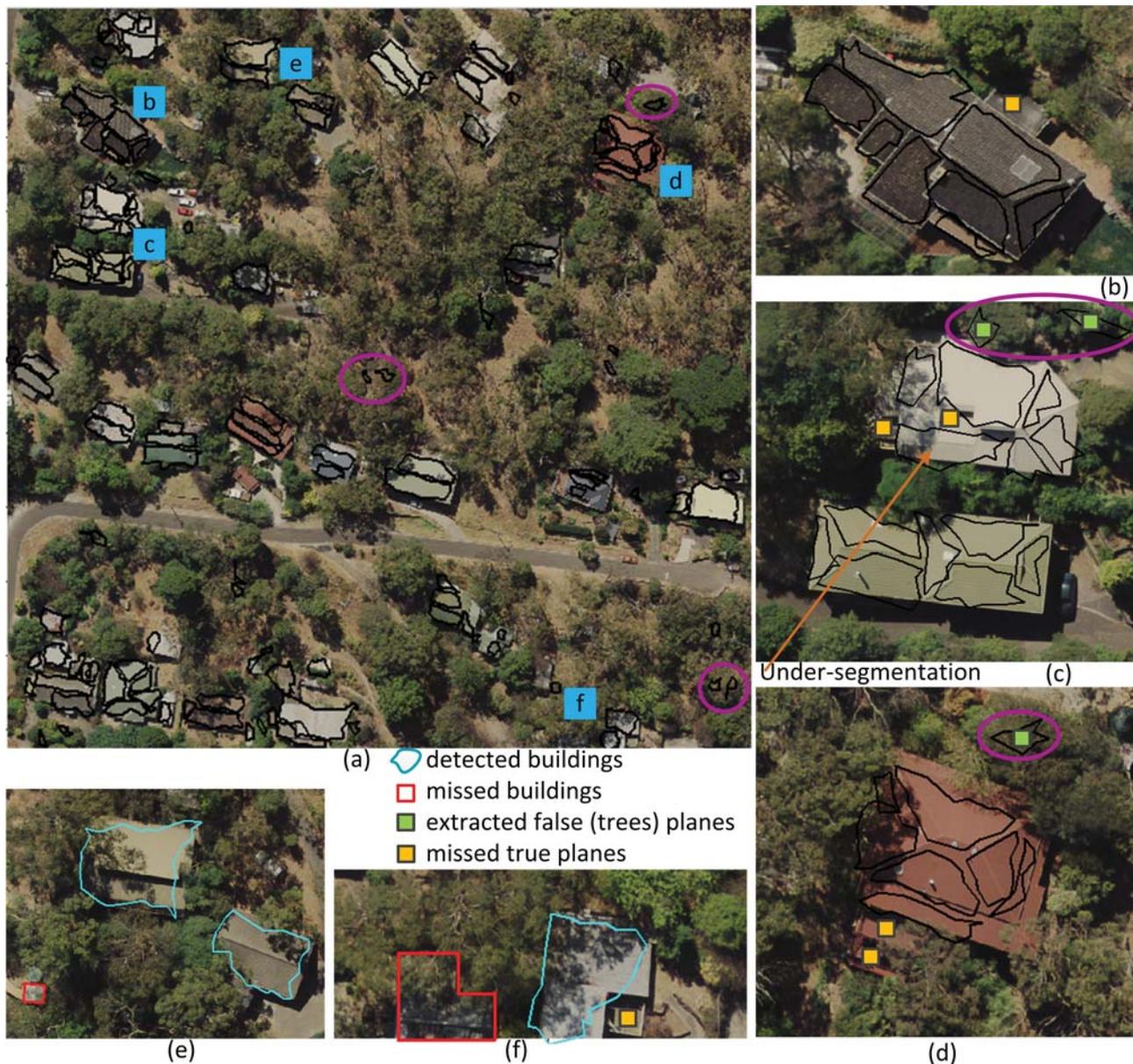
In pixel-based evaluation (Table 3), performance was worse in the HT and KN data sets than in the AV and HB data sets, due to the detection of some large trees and the missing of some small and occluded buildings. This is also evident from the high area omission and commission errors, as well as from the branching and miss factors for these two data sets. The accuracy of the extracted building boundaries is one to two times the maximum point spacing in the input LIDAR data. A better planimetric accuracy is shown to be possible with high density LIDAR data.

The detection cross-lap (under-segmentation) and reference cross-lap (over-segmentation) rates were high for the EL, KN and HT data sets, mainly due to dense vegetation in between neighbouring buildings and garage or garden sheds that were very close to buildings. For instance, two buildings in Figure 9d are detected jointly, because of the dense vegetation in between the buildings. The garden shed close to one of these buildings was merged with the building (Figure 8d). In the EL data set, there was one reference cross-lap, as well. As shown in Figure 8d, a part of a building is extracted separately. Because of the transparent material on the roof, there were laser returns from the ground (in between the building-part and the main building). Consequently, the building-part was found as a separate object in the building mask.

Some building detection examples in complex cases are shown in Figure 8b–d for the EL and in Figure 10e–f for the KN data sets. Thus, it has been demonstrated that the proposed method can extract buildings with complex roof structures, even when they are severely occluded by the surrounding trees. Nevertheless, the proposed method failed to detect small garden sheds, which are mostly occluded by trees, as shown by red polygons in Figures 8e–g and 10e–f.

Overall, as shown in Tables 2 and 3, the performance of the proposed method decreases with a decrease in LIDAR point density, and the worst performance was observed in the KN data set, where the LIDAR point density was only one point/m<sup>2</sup>. Most of the buildings missed in the KN data set were small in size and largely occluded by surrounding trees (see Figure 10e–f).

**Figure 10.** Roof plane extraction for (a) the Knox data set (one point/m<sup>2</sup>); (b–d) some extracted planes in complex cases; and (e–f) some extracted and missing buildings in difficult cases. Areas marked by letters in (a) are magnified in (b–f).



### 4.3.2. Roof Plane Extraction Results

Tables 6 and 7 show the object- and pixel-based evaluation results, respectively, for roof plane extraction for the VH data set using the evaluation system of [39]. Tables 8 and 9 show the same for other data sets, which were evaluated using the evaluation system of [41]. Figure 10 shows the roof plane extraction results for the KN data set. It also shows some samples for building detection and plane extraction results from this data set. Figure 11 shows the results for Area 2 of the VH data set and for the HB data set.

In the VH data set, the proposed algorithm performed better on Area 3, which comprises mainly residential buildings and less vegetation. For all three areas, there were many under-segmentation cases,

where small roof structures could not be separately extracted, but could be merged with the neighbouring large planes (see Figure 11c,d). In addition, as shown in Figure 11c, some low height roof structures were missed.

**Table 6.** Roof plane extraction results: object-based evaluation for the Vaihingen (VH) data set.  $C_m$  = completeness,  $C_r$  = correctness and  $Q_l$  = quality ( $C_{m,10}$ ,  $C_{r,10}$ ,  $Q_{l,10}$  are for planes over 10 m<sup>2</sup>) in percentage; 1: $M$  = over-segmentation and  $N$ :1 = under-segmentation;  $N$ : $M$  = both over- and under-segmentation in the number of planes.

Areas	$C_m$	$C_r$	$Q_l$	$C_{m,10}$	$C_{r,10}$	$Q_{l,10}$	1: $M$	$N$ :1	$N$ : $M$
VH 1	76.4	83.3	66.3	84.4	84.9	73.3	6	42	7
VH 2	73.9	91.9	69.4	93.8	92.6	87.2	7	3	1
VH 3	82.1	93.9	78	92.7	96.7	89.9	5	45	0
<b>Average</b>	<b>77.5</b>	<b>89.7</b>	<b>71.2</b>	<b>90.3</b>	<b>91.4</b>	<b>83.5</b>	<b>6</b>	<b>30</b>	<b>2.7</b>

**Table 7.** Roof plane extraction results: pixel-based and geometric evaluation for the Vaihingen (VH) data set.  $C_{mp}$  = completeness,  $C_{rp}$  = correctness and  $Q_{lp}$  = quality in percentage;  $RMSE$  = planimetric accuracy and  $RMSZ$  = height accuracy in metres.

Areas	$C_{mp}$	$C_{rp}$	$Q_{lp}$	$RMSE$	$RMSZ$
VH 1	90.5	91.9	83.8	1.05	0.41
VH 2	88.1	95.7	84.8	0.74	0.37
VH 3	91.5	91.9	84.7	0.89	0.27
<b>Average</b>	<b>90</b>	<b>93.2</b>	<b>84.4</b>	<b>0.89</b>	<b>0.35</b>

**Table 8.** Roof plane extraction results: object-based evaluation for the Aitkenvale (AV), Hervey Bay (HB), Eltham (EL), Hobart (HT) and Knox (KN) data sets in percentage.  $C_m$  = completeness,  $C_r$  = correctness and  $Q_l$  = quality ( $C_{m,10}$ ,  $C_{r,10}$ ,  $Q_{l,10}$  are for buildings over 10 m<sup>2</sup>);  $C_{rd}$  = detection cross-lap (under-segmentation) and  $C_{rr}$  = reference cross-lap (over-segmentation) rates.

Areas	$C_m$	$C_r$	$Q_l$	$C_{m,10}$	$C_{r,10}$	$Q_{l,10}$	$C_{rd}$	$C_{rr}$
AV 1	92.3	92.3	85.7	96	92.3	88.9	7.7	7.7
AV 2	68.6	94.8	66.1	85.9	94.8	82	16.9	4.1
HB	89.8	95.5	86.1	99.3	95.5	94.9	7.1	3
EL	79.1	87.9	71.4	92.6	87.9	82.1	10.9	5.4
HT	69.4	69.6	53.3	91	69.6	65.2	14.4	6.2
KN	56.4	59.3	40.6	76.7	59.3	50.2	23.4	3.3
<b>Average</b>	<b>75.9</b>	<b>83.2</b>	<b>67.2</b>	<b>90.3</b>	<b>83.2</b>	<b>77.2</b>	<b>13.4</b>	<b>5.0</b>

**Table 9.** Roof plane extraction results: pixel-based and geometric evaluation for the Aitkenvale (AV), Hervey Bay (HB), Eltham (EL), Hobart (HT) and Knox (KN) data sets.  $C_{mp}$  = completeness,  $C_{rp}$  = correctness and  $Q_{lp}$  = quality,  $A_{oe}$  = area omission error and  $A_{ce}$  = area commission error in percentage;  $RMSE$  = planimetric and  $RMSZ$  = height accuracies in metres.

Areas	$C_{mp}$	$C_{rp}$	$Q_{lp}$	$A_{oe}$	$A_{ce}$	$RMSE$	$RMSZ$
AV 1	85.6	80.8	71.1	14.4	19.2	0.49	0.032
AV 2	79.4	79.7	66.1	20.6	20.3	0.54	0.034
HB	85.5	72.1	64.2	14.5	27.9	0.52	0.029
EL	80.2	70.1	59.8	19.8	29.9	0.88	0.037
HT	72	59.2	48.1	28	40.8	1.00	0.042
KN	63.6	62.5	46.1	36.4	37.5	1.11	0.047
<b>Average</b>	<b>77.7</b>	<b>70.7</b>	<b>59.2</b>	<b>22.3</b>	<b>29.3</b>	<b>0.76</b>	<b>0.037</b>

Improved roof extraction results were observed in the AV and HB data sets (see Tables 8 and 9), where the LIDAR point density was high. The under-segmentation cases were higher than the over-segmentation cases, since some of the small planes were merged with neighbouring large planes. The high area omission and commission errors in Table 9 were partially because of the misalignment between the reference data and the roof plane extraction results. The reason for the misregistration is that in the case of the AV and HB data sets, the reference buildings and planes were generated from orthoimages using Barista software [44]. However, there were large registration errors (one to 2 m<sup>2</sup>) between the orthoimagery and LIDAR data. Since the proposed algorithm extracted information from LIDAR data only, the estimated omission and commission errors might include the registration error. Similar to the building detection results presented above, the planimetric accuracy in roof plane extraction was within one to two times the point spacing of the LIDAR data. The height error is only three to 5 cm, which shows the advantage of using raw LIDAR data, instead of a raster DSM.

As can be seen in the last two columns of Table 8, there were many under- (detection cross-lap) and over-segmentation (reference cross-lap) errors among the extracted roof planes. The under-segmentation error was the highest for the KN data set, where many of the small planes were either missed or merged with the neighbouring large planes. For instance, examples of missing roof planes are shown in Figure 10b–d. One of the extracted planes in Figure 9c covered at least two neighbouring planes (under-segmentation). In pixel-based evaluation (Table 9), the proposed method performed more poorly in the EL, HT and KN data sets than in the AV 1 and HB test areas, which were highly vegetated and hilly. This performance is indicated not only by lower completeness and correctness values, but also by high area omission and commission errors. The method performed the worst in the KN data set, which had the lowest point density.

Similar to the building detection performance, the overall roof extraction performance of the proposed method deteriorates with a decrease in LIDAR point density. Its performance is significantly affected at low point densities, for example, in the HT and KN data sets, where the point density was only one to two points/m<sup>2</sup>.

**Figure 11.** Roof plane extraction on (a) Area 2 of the VH data set and (b) the HB data set. Roof plane extraction examples in: (c–e) Area 2 of the VH data set and (f–g) the HB data set. Areas marked by letters in (a) and (b) are magnified in (c–g).



#### 4.4. Computation Time

Table 10 shows the computation time in seconds for each data set. Since the test areas differ in size and point density, both run time per  $m^2$  (the total time divided by the corresponding test area) and run

time per point (the total time divided by the number of non-ground points) are shown along with the total run time. The ground points were not considered, since they are not used at all after LIDAR classification and mask generation in Section 3.1. Computations were performed on a Windows 7 64-bit machine with an Intel(R) Xeon(R) CPU (E31245 @ 3.30 GHz) and 16 GB RAM.

**Table 10.** Computation time in seconds. Data sets: Vaihingen (VH), Aitkenvale (AV), Hervey Bay (HB), Eltham (EL), Hobart (HT) and Knox (KN).  $Time$  = total execution time;  $Time_{area}$  = time/m<sup>2</sup>; and  $Time_{point}$  = time/point.

Areas	$Time$	$Time_{area}$	$Time_{point}$
VH 1	301.3	0.00826	0.00516
VH 2	536.0	0.01143	0.00667
VH 3	322.2	0.00820	0.00550
AV 1	39.8	0.01100	0.00082
AV 2	5350.0	0.15723	0.00710
HB	81.8	0.00722	0.00157
EL	1074.6	0.01221	0.00430
HT	550.6	0.00602	0.00754
KN	268.2	0.00641	0.00722

In general, when the total run time in Table 10 is considered, the efficiency of the proposed method varies with respect to the point density, test area size and amount of vegetation. Area 2 of the VH data set required more time than the other two areas of the same data set, since it is larger and possesses more vegetation. The AV 2 area took the maximum time, because of its high point density and dense vegetation over a large area (see Figure 9b). The run time for the EL data set was second to the AV 2, due its dense vegetation over a large area, as shown in Figure 8. The AV 1 and HB areas took the least time, because they are small in size and have limited vegetation.

In terms of run time per m<sup>2</sup>, the three areas of the VH data set and the EL data set showed similar performance, since their point density was similar. In spite of having high point density, the AV 1 and HB areas took similar computation times to the VH and EL areas, since a significant parts of these two areas comprised ground. Area 2 of the AV data set was again the slowest, since it has high point density and dense vegetation. The HT and KN areas took the least time, because of their low point density.

As far as the run time per point is concerned, all test areas took 0.004 to 0.008 s, except the AV 1 and HB areas run time was shorter, because of the large areas of ground with limited vegetation.

#### 4.5. Comparison with Other Methods

Since the proposed algorithm is an automatic method and works solely with LIDAR data, the existing methods that use LIDAR data, shown in Table 11, have been considered for comparison. The results of the existing methods on the VH data set, shown in Tables 12 and 13, are available in [20,35,40]. The first three methods in Table 11 are used to compare building detection results, and the remaining two and Dorninger [26] are used to compare roof plane extraction results to those of the proposed method.

In all three VH areas, the proposed method of building detection offered similar or better completeness in both object- and pixel-based evaluation than the alternative methods, namely Dorninger, Niemeyer [35] and Liu [20], except in Areas 2 and 3, in which Niemeyer gave slightly better pixel- and object-based completeness, respectively, than the proposed method (see Table 12). Nevertheless, in terms of object-based correctness, while Niemeyer showed significantly worse results, Dorninger provided the best performance in all three areas. All three existing methods offered better pixel-based correctness than the proposed method. For buildings larger than 50 m<sup>2</sup>, while Dorninger and Liu were unable to detect some buildings, especially in Areas 1 and 3, Niemeyer and the proposed method were successful in detecting the majority of them. However, Niemeyer still could not remove some large trees in Areas 1 and 2, as shown by low  $C_{r,50}$  values in Table 12. In terms of planimetric accuracy, in all three areas, the proposed method produced better performance than Liu, but slightly worse results than Niemeyer.

**Table 11.** Existing methods that have been compared with the proposed method.

Method's Name	Reference
Niemeyer	Niemeyer <i>et al.</i> [35]
Dorninger	Dorninger and Pfeifer [26]
Liu	Liu <i>et al.</i> [20]
Oude Elberink	Oude Elberink and Vosselman [30]
Sohn	Sohn <i>et al.</i> [13]

**Table 12.** Comparing building detection results for the Vaihingen (VH) data set. Object-based  $C_m$  = completeness and  $C_r$  = correctness ( $C_{m,50}$  and  $C_{r,50}$  are for buildings over 50 m<sup>2</sup>) and pixel-based  $C_{mp}$  = completeness and  $C_{rp}$  = correctness are in percentage;  $RMSE$  = planimetric accuracy in metres. Results for existing methods are from [40].

Methods	$C_m$	$C_r$	$C_{m,50}$	$C_{r,50}$	$C_{mp}$	$C_{rp}$	$RMSE$
<b>Area 1</b>							
Niemeyer	83.8	72.1	100	87.9	87	90.1	1.09
Dorninger	78.4	100	96.7	100	85.7	98.1	0.99
Liu	75.7	93.5	93.3	96.7	76.7	95.7	1.18
Proposed	83.8	96.9	100	100	92.7	88.7	1.11
<b>Area 2</b>							
Niemeyer	78.6	52.4	100	84.6	93.8	91.4	0.71
Dorninger	85.7	100	100	100	85.4	98.4	1.17
Liu	71.4	100	90.9	100	88.5	98.9	0.71
Proposed	85.7	84.6	100	100	91.5	91	0.83
<b>Area 3</b>							
Niemeyer	82.1	90.2	97.5	100	93.8	93.7	0.65
Dorninger	75	100	95	100	86.3	98.7	0.81
Liu	55.4	100	77.5	100	67.8	98.4	1.17
Proposed	78.6	97.8	97.4	100	93.9	86.3	0.89

**Table 13.** Comparing roof plane extraction results for the Vaihingen (VH) data set. Object-based  $C_m$  = completeness and  $C_r$  = correctness ( $C_{m,10}$  and  $C_{r,10}$  are for planes over 10 m<sup>2</sup>) in percentage; 1: $M$  = over-segmentation and  $N$ :1 = under-segmentation,  $N$ : $M$  = both over- and under-segmentation in the number of planes;  $RMSE$  = planimetric accuracy and  $RMSZ$  = height accuracy in metres. Results for existing methods are from [40].

Methods	$C_m$	$C_r$	$C_{m,10}$	$C_{r,10}$	1: $M$	$N$ :1	$N$ : $M$	$RMSE$	$RMSZ$
<b>Area 1</b>									
Oude Elberink	65.3	97.3	63.3	97.3	0	38	3	0.94	0.55
Dorninger	72.2	96.7	77.7	96.5	7	42	6	0.79	0.65
Sohn	88.2	98.5	89.9	98.2	5	36	14	0.75	0.58
Proposed	76.4	83.3	84.4	84.9	6	42	7	1.05	0.41
<b>Area 2</b>									
Oude Elberink	79.7	95	94	100	0	7	0	1.16	3.31
Dorninger	73.9	100	88	100	3	5	1	1.03	0.88
Sohn	73.9	100	90	100	5	3	0	0.77	1.04
Proposed	73.9	91.9	93.8	92.6	7	3	1	0.74	0.37
<b>Area 3</b>									
Oude Elberink	64.3	100	55.9	100	0	46	0	1.04	0.42
Dorninger	76.6	99.1	74.5	99.1	3	50	0	0.84	0.38
Sohn	84.7	100	89	100	2	51	1	0.77	0.35
Proposed	82.1	93.9	92.7	96.7	5	45	0	0.89	0.27

In Areas 1 and 3 of the VH data set, the proposed roof plane extraction method offered better object-based completeness than Oude Elberink [30] (see Table 13). In Area 2, Oude Elberink showed better completeness than the proposed method, but as the plane size increased (>10 m<sup>2</sup>), both methods showed similar completeness. In terms of planimetric accuracy, the proposed algorithm performed better than Oude Elberink in Areas 2 and 3, but worse in Area 1.

A similar performance difference was found when the results were compared with those by Dorninger. In all three VH areas, in terms of object-based evaluation, the proposed method showed better completeness, but lower correctness, than Dorninger, even for the planes larger than 10 m<sup>2</sup>. The under-segmentation error was lower for the proposed method. Moreover, Dorninger is a semiautomatic method, as it applies manual preprocessing and post-processing steps.

In Area 2 of the VH data set, the proposed method offered better object-based completeness and height accuracy than Sohn [13]. However, in terms of completeness and correctness, Sohn outperformed the proposed method in the other two VH areas. In Area 1, the proposed method showed more under-segmentation errors, but a lower total number of over- and under-segmentation errors than Sohn. In Area 3, the method showed less under-segmentation errors than Sohn.

For the extracted planes in all VH areas, the proposed method showed much better height accuracy than Oude Elberink, Dorninger and Sohn, while Dorninger showed significantly larger height error (more than 3.3 m) in Area 2.

## 5. Conclusions

A new LIDAR point cloud segmentation algorithm has been proposed for automatic extraction of 3D building roof planes from LIDAR data. It has been shown via experimental testing that the proposed algorithm affords high building detection rates and good roof plane extraction performance. It is not only capable of detecting small buildings, but can also extract small roof planes on complex building roofs. Moreover, in most cases, it can separate buildings from surrounding dense vegetation.

However, since the method uses LIDAR data alone, the planimetric accuracy is limited by the LIDAR point density. At present, the method does not incorporate smoothing of the boundaries of extracted planar segments. Moreover, it will not work on curved roofs. Future work will look at the development of a regularisation procedure to smooth roof plane boundaries and to reconstruct building roof models. The integration of image data will also help for better object extraction where LIDAR information is missing. An extension of the proposed method could be the use of higher density point cloud data (low density LIDAR point cloud data may be complemented with DSMs derived from dense image matching [9]), and thus, the curved surfaces could be better approximated [26]. An alternative is to follow a hybrid approach [46] that can incorporate both data-driven and model-driven approaches. While the data-driven approach, presented in this paper, will aim to extract planar roof segments, the model-driven approach will concentrate on the extraction of curved roof surfaces.

## Acknowledgment

M. Awrangjeb is a recipient of the Discovery Early Career Researcher Award by the Australian Research Council (project number DE120101778). The Vaihingen data set was provided by the German Society for Photogrammetry, Remote Sensing and Geoinformation [36]. The Aitkenvale and Hervey Bay data sets were provided by Ergon Energy in Queensland, Australia. The Hobart data set was provided by Photomapping Services in Melbourne, Australia. The Knox and Eltham data sets were provided by the Department of Environment and Primary Industries of Victoria, Australia.

## Author Contributions

M. Awrangjeb has developed, implemented and conducted the tests on the benchmark and Australian data sets. In addition, he has written the paper. Clive S. Fraser has supervised M. Awrangjeb and reviewed the paper for organisation, clarification and English corrections.

## Conflicts of Interest

The authors declare no conflicts of interest.

## References

1. Zhang, Y.; Zhang, Z.; Zhang, J.; Wu, J. 3D building modelling with digital map, LIDAR data and video image sequences. *Photogramm. Record* **2005**, *20*, 285–302.
2. Cheng, L.; Gong, J.; Li, M.; Liu, Y. 3D building model reconstruction from multi-view aerial imagery and LIDAR data. *Photogramm. Eng. Remote Sens.* **2011**, *77*, 125–139.
3. Rau, J.Y.; Chen, L.C. Robust reconstruction of building models from three-dimensional line segments. *Photogramm. Eng. Remote Sens.* **2003**, *69*, 181–188.
4. Brenner, C. Towards fully automatic generation of city models. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2000**, *33*, 85–92.
5. Rottensteiner, F. Automatic generation of high-quality building models from LIDAR data. *IEEE Comput. Graph. Appl.* **2003**, *23*, 42–50.
6. Zhang, J.X.; Lin, X.G.; Ning, X.G. SVM-based classification of segmented airborne LIDAR point clouds in urban areas. *Remote Sens.* **2013**, *5*, 3749–3775.
7. Awrangjeb, M.; Zhang, C.; Fraser, C.S. Automatic extraction of building roofs using LIDAR data and multispectral imagery. *ISPRS J. Photogramm. Remote Sens.* **2013**, *83*, 1–18.
8. Khoshelham, K.; Li, Z.; King, B. A split-and-merge technique for automated reconstruction of roof planes. *Photogramm. Eng. Remote Sens.* **2005**, *71*, 855–862.
9. Gerke, M. Dense matching in high resolution oblique airborne images. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2009**, *XXXVIII*, 77–82.
10. Nex, F.; Rupnik, E.; Remondino, F. Building footprints extraction from oblique imagery. *ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci.* **2013**, *II*, 61–66.
11. 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.
12. Jochem, A.; Höfle, B.; Wichmann, V.; Rutzinger, M.; Zipf, A. Area-wide roof plane segmentation in airborne LIDAR point clouds. *Comput. Environ. Urban Syst.* **2012**, *36*, 54–64.
13. Sohn, G.; Huang, X.; Tao, V. Using binary space partitioning tree for reconstructing polyhedral building models from airborne LIDAR data. *Photogramm. Eng. Remote Sens.* **2008**, *74*, 1425–1438.
14. Awrangjeb, M.; Fraser, C.S. Rule-based segmentation of LIDAR point cloud for automatic extraction of building roof planes. *ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci.* **2013**, *II*, 1–6.
15. Sampath, A.; Shan, J. Segmentation and reconstruction of polyhedral building roofs from aerial LIDAR point clouds. *IEEE Trans. Geosci. Remote Sens.* **2010**, *48*, 1554–1567.
16. Awrangjeb, M.; Ravanbakhsh, M.; Fraser, C.S. Automatic detection of residential buildings using LIDAR data and multispectral imagery. *ISPRS J. Photogramm. Remote Sens.* **2010**, *65*, 457–467.
17. Novacheva, A. Building roof reconstruction from LIDAR data and aerial images through plane extraction and colour edge detection. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2008**, *37*, 53–58.
18. Laefer, D.F.; Hinks, T.; Carr, H.; Truong-Hong, L. New advances in automated urban model population through aerial laser scanning. *Recent Pat. Eng.* **2011**, *5*, 196–208.

19. Miliareisis, G.; Kokkas, N. Segmentation and object-based classification for the extraction of the building class from LIDAR DEMs. *Comput. Geosci.* **2007**, *33*, 1076–1087.
20. Liu, C.; Shi, B.; Xuan, X.; Nan, L. LEGION segmentation for building extraction from LIDAR data. *ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci.* **2012**, *34*, 291–296.
21. Cheng, L.; Zhao, W.; Han, P.; Zhang, W.; Shan, J.; Liu, Y.; Li, M. Building region derivation from LIDAR data using a reversed iterative mathematic morphological algorithm. *Opt. Commun.* **2013**, *286*, 244–250.
22. Dos Santos Galvanin, E.A.; Poz, A.P.D. Extraction of building roof contours from LIDAR data using a Markov-Random-Field-based approach. *IEEE Trans. Geosci. Remote Sens.* **2012**, *50*, 981–987.
23. Zhou, Q.Y.; Neumann, U. Fast and Extensible Building Modeling from Airborne LIDAR Data. In Proceedings of the 16th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, Irvine, CA, USA, 5–7 November 2008; pp. 1–8.
24. Perera, S.N.; Nalani, H.A.; Maas, H.G. An automated method for 3D roof outline generation and regularization in airborne laser scanner data. *ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci.* **2012**, *1–3*, 281–286.
25. Vosselman, G.; Gorte, B.G.H.; Sithole, G.; Rabbani, T. Recognising structure in laser scanner point cloud. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2004**, *36*, 33–38.
26. Dorninger, P.; Pfeifer, N. A comprehensive automated 3D approach for building extraction, reconstruction, and regularization from airborne laser scanning point clouds. *Sensors* **2008**, *8*, 7323–7343.
27. Verma, V.; Kumar, R.; Hsu, S. 3D Building Detection and Modeling from Aerial LIDAR Data. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, New York, NY, USA, 17–22 June 2006; pp. 2213–2220.
28. Kim, K.; Shan, J. Building roof modeling from airborne laser scanning data based on level set approach. *ISPRS J. Photogramm. Remote Sens.* **2011**, *66*, 484–497.
29. Sun, S.; Salvaggio, C. Aerial 3D building detection and modeling from airborne LIDAR point clouds. *IEEE J. Sel. Top. Appl. Earth Observ. Remote Sens.* **2013**, *6*, 1440–1449.
30. Oude Elberink, S.O.; Vosselman, G. Building reconstruction by target based graph matching on incomplete laser data: Analysis and limitations. *Geodetski Vestnik* **2009**, *9*, 6101–6118.
31. Tarsha-Kurdi, F.; Landes, T.; Grussenmeyer, P. Extended RANSAC algorithm for automatic detection of building roof planes from LIDAR data. *Photogramm. J. Finl.* **2008**, *21*, 97–109.
32. Richter, R.; Behrens, M.; Döllner, J. Object class segmentation of massive 3D point clouds of urban areas using point cloud topology. *Int. J. Remote Sens.* **2013**, *34*, 8408–8424.
33. Carlberg, M.; Gao, P.; Chen, G.; Zakhor, A. Classifying Urban Landscape in Aerial LIDAR Using 3D Shape Analysis. In Proceedings of the IEEE International Conference on Image Processing, Cairo, Egypt, 7–10 November 2009; pp. 1701–1704.
34. Zhou, Q.Y.; Neumann, U. Complete residential urban area reconstruction from dense aerial LIDAR point clouds. *Graph. Models* **2013**, *75*, 118–125.

35. Niemeyer, J.; Wegner, J.D.; Mallet, C.; Rottensteiner, F.; Soergel, U. Conditional Random Fields for Urban Scene Classification with Full Waveform LIDAR Data. In *Photogrammetric Image Analysis*; Stilla, U., Rottensteiner, F., Mayer, H., Jutzi, B., Butenuth, M., Eds.; Springer Berlin Heidelberg: Berlin, Germany, 2011; pp. 233–244.
36. Cramer, M. The DGPF test on digital aerial camera evaluation-overview and test design. *Photogramm. Fernerkundung Geoinf.* **2010**, *2*, 73–82.
37. Rottensteiner, F.; Trinder, J.; Clode, S.; Kubik, K. Building Detection Using LIDAR Data and Multispectral Images. In Proceedings of the Digital Image Computing: Techniques and Applications, Sydney, Australia, 10–12 Decemeber 2003; pp. 673–682.
38. Awrangjeb, M.; Zhang, C.; Fraser, C.S. Building detection in complex scenes thorough effective separation of buildings from trees. *Photogramm. Eng. Remote Sens.* **2012**, *78*, 729–745.
39. Rutzinger, M.; Rottensteiner, F.; Pfeifer, N. A comparison of evaluation techniques for building extraction from airborne laser scanning. *IEEE J. Sel. Top. Appl. Earth Observ. Remote Sens.* **2009**, *2*, 11–20.
40. Awrangjeb, M.; Fraser, C.S. Automatic and Threshold-Free Evaluation of 3D Building Roof Recosntruction Techniques. In Proceedings of the IEEE International Geoscience and Remote Sensing Symposium, Melbourne, Australia, 21–26 July 2013; pp. 3970–3973.
41. Rottensteiner, F.; Sohn, G.; Jung, J.; Gerke, M.; Baillard, C.; Benitez, S.; Breitkopf, U. The ISPRS benchmark on urban object classification and 3D building reconstruction. *ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci.* **2012**, *1–3*, 293–298.
42. Test Results. Available online: <http://www2.isprs.org/commissions/comm3/wg4/results.html> (accessed on 15 January 2014).
43. Rottensteiner, F.; Sohn, G.; Gerke, M.; Wegner, J.D.; Breitkopf, U.; Jung, J. Results of the ISPRS benchmark on urban object detection and 3D building reconstruction. *ISPRS J. Photogramm. Remote Sens.* **2013**, *2013*, doi:10.1016/j.isprsjprs.2013.10.004.
44. The Barista Software. Available online: [www.baristasoftware.com.au](http://www.baristasoftware.com.au) (accessed on 15 January 2014).
45. Roof Extraction Results and Data Sets. Available online: <http://users.monash.edu.au/~mawrangj/RExtraction.html> (accessed on 19 April 2014).
46. Satari, M.; Samadzadegan, F.; Azizi, A.; Maas, H.G. A Multi-resolution hybrid approach for building model reconstruction from LIDAR data. *Photogramm. Record* **2012**, *27*, 330–359.