ISPRS Journal of Photogrammetry and Remote Sensing 83 (2013) 1–18

Contents lists available at SciVerse ScienceDirect

ISPRS Journal of Photogrammetry and Remote Sensing

journal homepage: www.elsevier.com/locate/isprsjprs

Automatic extraction of building roofs using LIDAR data and multispectral imagery

Mohammad Awrangjeb [a,*], Chunsun Zhang [b], Clive S. Fraser [b]

[a] Gippsland School of Information Technology, Monash University, Vic 3842, Australia
[b] CRC for Spatial Information, Dept. of Infrastructure Engineering, University of Melbourne, Australia

ARTICLE INFO

Article history:
Received 22 January 2013
Received in revised form 14 May 2013
Accepted 14 May 2013

Keywords:
Building
Feature
Extraction
Reconstruction
Automation
Integration
LIDAR
Orthoimage

ABSTRACT

Automatic 3D extraction of building roofs from remotely sensed data is important for many applications including city modelling. This paper proposes a new method for automatic 3D roof extraction through an effective integration of LIDAR (Light Detection And Ranging) data and multispectral orthoimagery. 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 are exploited to constitute a 'ground mask'. The second group contains the non-ground points which are segmented using an innovative image line guided segmentation technique to extract the roof planes. The image lines are extracted from the grey-scale version of the orthoimage and then classified into several classes such as 'ground', 'tree', 'roof edge' and 'roof ridge' using the ground mask and colour and texture information from the orthoimagery. During segmentation of the non-ground LIDAR points, the lines from the latter two classes are used as baselines to locate the nearby LIDAR points of the neighbouring planes. For each plane a robust seed region is thereby defined using the nearby non-ground LIDAR points of a baseline and this region is iteratively grown to extract the complete roof plane. Finally, a newly proposed rule-based procedure is applied to remove planes constructed on trees. Experimental results show that the proposed method can successfully remove vegetation and so offers high extraction rates.

1. Introduction

Up to date 3D building models are important for many GIS (Geographic Information System) applications such as urban planning, disaster management and automatic city planning (Gröger and Plümer, 2012). Therefore, 3D building reconstruction has been an area of active research within the photogrammetric, remote sensing and computer vision communities for the last two decades. Building reconstruction implies the extraction of 3D building information, which includes corners, edges and planes of the building facades and roofs from remotely sensed data such as aerial imagery and LIDAR (Light Detection And Ranging) data. The facades and roofs are then reconstructed using the available information. Although the problem is well understood and in many cases accurate modelling results are delivered, the major drawback is that the current level of automation is comparatively low (Cheng et al., 2011).

Three-dimensional building roof reconstruction from aerial imagery alone seriously lacks in automation partially due to shadows, occlusions and poor contrast. The introduction of LIDAR has offered a favourable option for improving the level of automation in 3D reconstruction when compared to image-based reconstruction alone. However, the quality of the reconstructed building roofs from LIDAR data is restricted by the ground resolution of the LIDAR which is still generally lower than that of the aerial imagery. That is why the integration of aerial imagery and LIDAR data has been considered complementary in automatic 3D reconstruction of building roofs. The issue of how to optimally integrate data from the two sources with dissimilar characteristics is still to be resolved and relatively few approaches have thus far been published.

Different approaches for building roof reconstruction have been reported in the literature. In the *model driven* approach, also known as the *parametric* approach, a predefined catalogue of roof forms (e.g., flat, saddle, etc.) is prescribed and the model that best fits the data is chosen. An advantage of this approach is that the final roof shape is always topologically correct. The disadvantage, however, is that complex roof shapes cannot be reconstructed if they are not in the input catalogue. In addition, the level of detail in the reconstructed building is compromised as the input models

* Corresponding author. Tel.: +61 3 990 26462; fax: +61 3 9902 7137.
E-mail addresses: mohammad.awrangjeb@monash.edu (M. Awrangjeb), chunsunz@unimelb.edu.au (C. Zhang), c.fraser@unimelb.edu.au (C.S. Fraser).
URL: http://users.monash.edu.au/mawrangj/ (M. Awrangjeb).

0924-2716/$ - see front matter © 2013 International Society for Photogrammetry and Remote Sensing, Inc. (ISPRS) Published by Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.isprsjprs.2013.05.006

usually consist of rectangular footprints. In the *data driven* approach, also known as the *generic* approach (Lafarge et al., 2010) or polyhedral approach (Satari et al., 2012), the roof is reconstructed from planar patches derived from segmentation algorithms. The challenge here is to identify neighbouring planar segments and their relationship, for example, coplanar patches, intersection lines or step edges between neighbouring planes. The main advantage of this approach is that polyhedral buildings of arbitrary shape may be reconstructed (Rottensteiner, 2003). The main drawback of data driven methods is their susceptibility to the incompleteness and inaccuracy of the input data; for example, low contrast and shadow in images and low point density in LIDAR data. Therefore, some roof features such as small dormer windows and chimneys cannot be represented if the resolution of the input data is low. Moreover, if a roof is assumed to be a combination of a set of 2D planar faces, a building with a curved roof structure cannot be reconstructed. Nonetheless, in the presence of high density LIDAR and image data, curved surfaces can be well approximated (Dorninger and Pfeifer, 2008). The *structural* approach, also known as the *global strategy* (Lafarge et al., 2010) or Hybrid approach (Satari et al., 2012), exhibits both model and data driven characteristics. For example, Satari et al. (2012) applied the data driven approach to reconstruct cardinal planes and the model-driven approach to reconstruct dormers.

The reported research in this paper concentrates on 3D extraction of roof planes. A new data driven approach is proposed for automatic 3D roof extraction through an effective integration of LIDAR data and multispectral imagery. The LIDAR data is divided into two groups: ground and non-ground points. The ground points are used to generate a 'ground mask'. The non-ground points are iteratively segmented to extract the roof planes. The structural image lines are classified into several classes ('ground', 'tree', 'roof edge' and 'roof ridge') using the ground mask, colour orthoimagery and image texture information. In an iterative procedure, the non-ground LIDAR points near to a long roof edge or ridge line (known as the baseline) are used to obtain a roof plane. Finally, a newly proposed rule-based procedure is applied to remove planes constructed on trees. Promising experimental results for 3D extraction of building roofs have been obtained for two test data sets.

Note that the initial version of this method was introduced in Awrangjeb et al. (2012a), where the preliminary idea was briefly presented without any objective evaluation of the extracted roof planes. This paper not only presents full details of the approach and the objective evaluation results, but also proposes a new rule-based procedure in order to remove trees.

The rest of the paper is organised as follows: Section 2 presents a review of the prominent data driven methods for 3D building roof extraction. Section 3 details the proposed extraction algorithm. Section 4 presents the results for two test data sets, discusses the sensitivity of two algorithmic parameters and compares the results of the proposed technique with those of existing data driven techniques. Concluding remarks are then provided in Section 5.

2. Literature review

The 3D reconstruction of building roofs comprises two important steps (Rottensteiner et al., 2004). The *detection* step is a classification task and delivers regions of interest in the form of 2D lines or positions of the building boundary. The *reconstruction* step constructs the 3D models within the regions of interest using the available information from the sensor data. The detection step significantly reduces the search space for the reconstruction step. In this section, a review of some of the prominent *data driven* methods for 3D roof reconstruction is presented.

Methods using ground plans (Vosselman and Dijkman, 2001) simplify the problem by partitioning the given plan and finding the most appropriate planar segment for each partition. However, in the absence of a ground plan or if it is not up to date, such methods revert to semi-automatic (Dorninger and Pfeifer, 2008). Rottensteiner (2003) automatically generated 3D building models from point clouds alone. However, due to the use of LIDAR data alone, the level of detail of the reconstructed models and their positional accuracy were poor. An improvement involving the fusion of high resolution aerial imagery with a LIDAR DSM (Digital Surface Model) was latter proposed (Rottensteiner et al., 2004).

Khoshelham et al. (2005) applied a split-and-merge technique on a DSM guided image segmentation technique for automatic extraction of roof planes. In evaluation, the accuracy of reconstructed planes was shown for four simple gable roofs only. Chen et al. (2006) reconstructed buildings with straight (flat and gable roofs only) and curvilinear (flat roof only) boundaries from LIDAR and image data. Though the evaluation results were promising, the method could not detect buildings smaller than 30 m^2 in area and for the detected buildings both planimetric and height errors were high.

Park et al. (2006) reconstructed large complex buildings using LIDAR data and digital maps. Unlike other methods, this method was able to reconstruct buildings as small as 4 m^2. However, in the absence of a ground plan, or if the plan is not up to date, the method becomes semi-automatic. In addition, objective evaluation results were missing in the published paper. Dorninger and Pfeifer (2008) proposed a method using LIDAR point clouds. Since the success of the proposed automated procedure was low, the authors advised manual pre-processing and post-processing steps. In the pre-processing step, a coarse selection of building regions was accomplished by digitizing each building interactively. In the post-processing step, the erroneous building models were indicated and rectified by means of commercial CAD software. Moreover, some of the algorithmic parameters were set interactively. Sampath and Shan (2010) presented a solution framework for segmentation (detection) and reconstruction of polyhedral building roofs from high density LIDAR data. They provided good evaluation results for both segmentation and reconstruction. However, due to removal of LIDAR points near the plane boundaries, the method exhibited high reconstruction errors on small planes. Furthermore, the fuzzy k-means clustering algorithm was computationally expensive (Khoshelham et al., 2005).

Habib et al. (2010) reported on semi-automatic polyhedral building model generation through integration of LIDAR data and stereo imagery. Planar roof patches were first generated from the LIDAR data and then 3D image lines were matched along the LIDAR boundaries. Finally, a manual monoplotting procedure was used to both delete incorrect boundaries and add necessary boundary segments. Some true boundaries were missed and erroneous boundaries were detected due to relief displacement, shadows and low image contrast. Cheng et al. (2011) integrated multi-view aerial imagery with LIDAR data for 3D building model reconstruction. This was a semi-automatic method since in many cases 20–30% of roof lines needed to be manually edited. In addition, this method was computationally expensive and failed to reconstruct complex roof structures. Jochem et al. (2012) proposed a roof plane segmentation technique from raster LIDAR data using a seed point based region growing technique. Vegetation was removed using the slope-adaptive LIDAR echo ratio and the approach showed good object-based evaluation results on a large data set using a threshold-free evaluation system. However, because of the use of gridded height data, there was an associated loss of accuracy in the extracted planes.

3. Proposed extraction procedure

Fig. 1 shows an overview of the proposed building roof extraction procedure. The input data consists of raw LIDAR data and multispectral or colour orthoimagery. In the detection step (top dashed rectangle in Fig. 1), the LIDAR points on the buildings and trees are separated as non-ground points. The primary building mask known as the 'ground mask' (Awrangjeb et al., 2010b) is generated using the LIDAR points on the ground. The NDVI (Normalised Difference Vegetation Index) is calculated for each image pixel location using the multispectral orthoimage. If multispectral orthoimagery is not available, then the pseudo-NDVI is calculated from a colour orthoimage. From this point forward, the term NDVI is used for both indices.

Texture information like entropy is estimated at each image pixel location using a grey-scale version of the image (Gonzalez et al., 2003). The same grey level image is used to find lines in the image that are at least 1 m in length (Awrangjeb and Lu, 2008). These lines are classified into several classes, namely, 'ground', 'tree', 'roof edge' (roof boundary) and 'roof ridge' (intersection of two roof planes) using the ground mask, NDVI and entropy information (Awrangjeb et al., 2012b). In the extraction step (bottom dashed rectangle in Fig. 1), lines classified as roof edges and ridges are processed along with the non-ground LIDAR points. During LIDAR plane extraction, LIDAR points near to a roof edge or ridge, which is considered as the baseline for the plane, are used to start the newly proposed region growing algorithm. LIDAR points that are compatible with the plane are then iteratively included as part of the plane. Other planes on the same building are extracted following the same procedure by using the non-ground LIDAR points near to local image lines. Finally, the false positive planes, mainly constructed on trees, are removed using information such as size, and spikes within the extracted plane boundaries. The remaining planes form the final output.

In order to obtain a planar roof segment from the LIDAR data, it is usual to initially choose a seed surface or seed region (Vosselman et al., 2004). Then the points around the seed are considered to iteratively grow the planar segment. Jiang and Bunke (1994) defined a seed region on three adjacent scan lines (or rows of a range image). This method cannot be used with raw LIDAR points. Others used a brute-force method for seed selection where they fit many planes and analyzed their residuals. This method is expensive and does not work well in the presence of outliers (Vosselman et al., 2004).

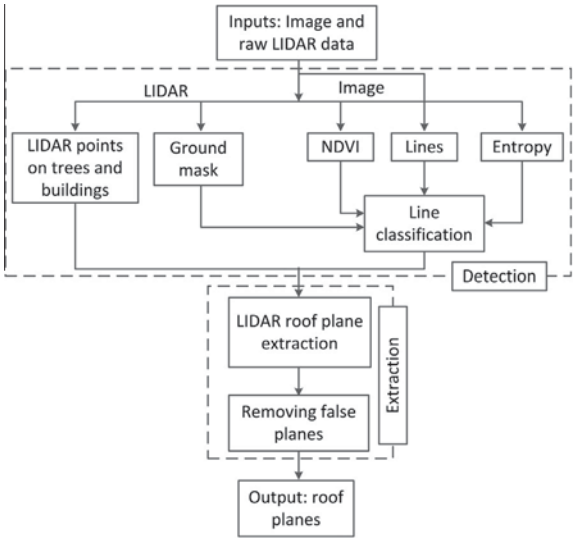

Fig. 1. Proposed method for automatic 3D extraction of building roofs.

When a building is considered as a polyhedral model, each of the planar segments on its roof corresponds to a part in the LIDAR data where all points within some distance belong to the same surface. If this part can be identified as a seed region, the brute-force method can be avoided. This paper proposes a method to define such a seed region for each roof plane with the help of image lines, and then to extend the seed region iteratively to complete the planar segment.

In the following sections, a sample of a data set used for experimentation is first presented, and then the detection and extraction steps of the proposed 3D roof extraction method are detailed.

3.1. Sample Test Data

Fig. 2a presents a sample scene from the test data set. It will be used to illustrate the different steps of the proposed extraction method. This scene is from Aitkenvale, Queensland, Australia. Available data comprised first-pulse LIDAR returns with a point spacing of 0.17 m (35 points/m^2, Fig. 2b) and an RGB colour orthoimage with a resolution of 0.05 m.

Although the image and LIDAR data were registered using a mutual information based technique (Parmehr et al., 2012), there were still significant misalignments between the two data sets, because the orthoimage had been created using a bare-earth DEM (Digital Elevation Model). The roofs and tree-tops were thus displaced considerably with respect to the LIDAR data and the alignment was not perfect. Apart from this registration problem, there were also problems with shadows in the orthoimage, so the NDVI image, shown in Fig. 2c, did not provide as much information as expected. Therefore, texture information in the form of entropy (Gonzalez et al., 2003) (see Fig. 2d) was also employed based on the observation that trees are rich in texture as compared to building roofs. While a high entropy value at an image pixel indicates a texture (tree) pixel, a low entropy value indicates a 'flat' (building roof) pixel. The entropy and NDVI information together will be used to identify the roof and tree edges while classifying image lines. However, as with the building detection algorithm reported by Awrangjeb et al. (2012b), the proposed extraction algorithm does not remove all trees at the initial stage. It employs a new rule-based procedure to remove false positive planes on trees (see Section 3.3.2). Note that the image entropy rather than the texture information from the LIDAR data (such as entropy and the difference between the first and last echo) has been used due to the high image resolution and unavailability of LIDAR data with two or multiple returns.

3.2. Roof line detection

In this section, the LIDAR classification, ground mask generation and image line extraction and classification procedures of the detection step shown in Fig. 1 are presented.

3.2.1. LIDAR classification and Mask generation

For each LIDAR point, the corresponding DEM height is used as the ground height H_g. A height threshold $T_h = H_g + 2.5$ m (Rottensteiner et al., 2004) is then applied to the raw LIDAR height. Consequently, the LIDAR data are divided into two groups: *ground points* such as ground, road furniture, cars and bushes which are below the threshold, and *non-ground points* which represent elevated objects such as buildings and trees.

Two masks – primary (from ground points) and secondary (from non-ground points) as shown in Fig. 3a and b – are generated following the procedure in Awrangjeb et al. (2010b, 2012b). The primary or ground mask M_g indicates the *void areas* where there

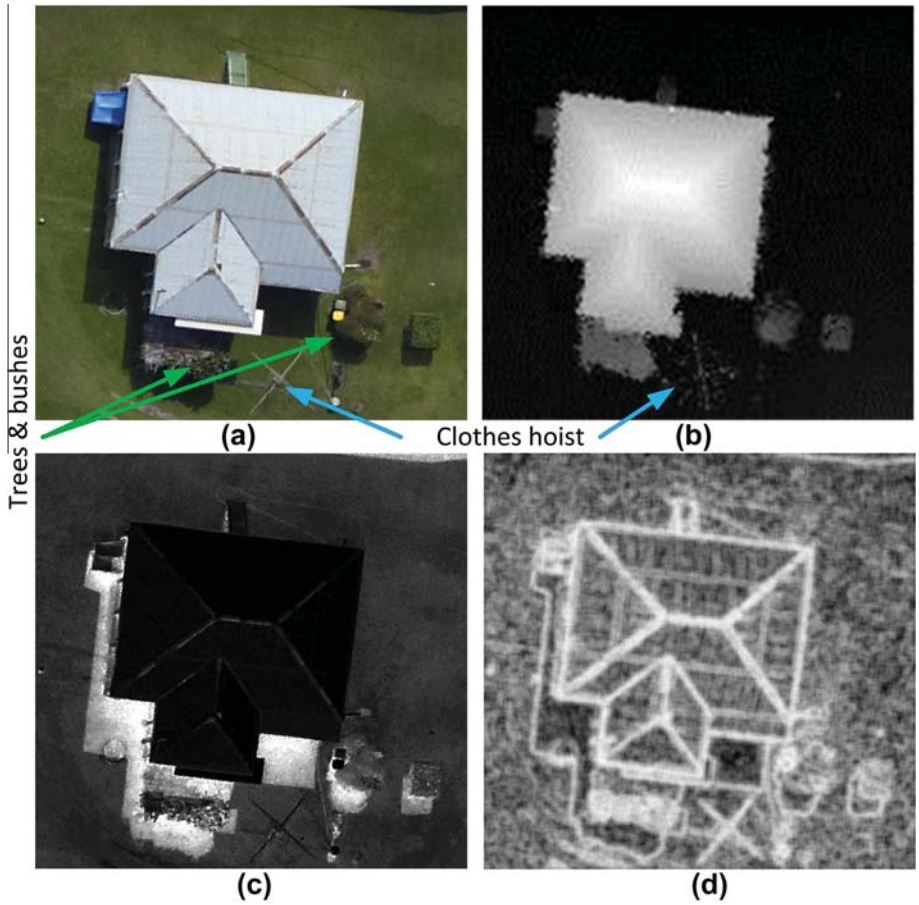

Fig. 2. Sample data: (a) RGB orthoimage, (b) LIDAR data shown as a grey-scale grid, (c) NDVI image from (a), and (d) entropy image from (a).

are no laser returns below T_h, ie, ground areas covered by buildings and trees. In contrast, the secondary or non-ground mask, which is equivalent to the normalized DSM (nDSM), indicates the *filled areas*, from where the laser reflects, above the same threshold, ie, roofs and tree tops. Awrangjeb et al. (2012b) has shown that buildings and trees are found to be thinner in M_g than in nDSM. This is also evident from Fig. 3a and b. While both the tree and bush near to the building are detected in nDSM, one of them is completely missed in M_g and the other is almost separated from the building. Moreover, in nDSM the outdoor clothes hoist is clearly detected, but it is missed in M_g. Consequently, M_g is used to classify image lines as discussed below.

3.2.2. Line extraction and classification

In order to extract lines from a grey-scale orthoimage, edges are first detected using the Canny edge detector. Corners are then detected on the extracted curves via a fast corner detector (Awrangjeb et al., 2009). On each edge, all the pixels between two corners, or between a corner and an endpoint or two endpoints when enough corners are not available, are considered to form a separate line segment. If a line segment is smaller than the minimum image line length $l_m = 1$ m, it is removed. Thus, trees having small horizontal areas are removed. Finally, a least-squares straight-line fitting technique is applied to properly align each of the remaining line segments. A more detailed description of the image line extraction procedure can be found in Awrangjeb and Lu (2008). Fig. 4a shows the extracted lines from the test scene.

Three types of information are required to classify the extracted image lines into 'ground', 'trees', 'roof ridge' and 'roof edge': the ground mask (Fig. 3a), NDVI (Fig. 2c) and the entropy mask. In order to derive the entropy mask, the entropy image shown in Fig. 2d is first estimated. The mask is then created by applying an entropy threshold (see Gonzalez et al. (2003) and Awrangjeb et al. (2012b) for more details). The entropy mask for the test sample is shown in Fig. 4b.

For classification of the extracted image lines, a rectangular area of width $w_d = \frac{W_m}{2}$ on each side of a line is considered, where $W_m = 3$ m is assumed to be the minimum building width. The rectangular neighborhood setup procedure is further described in Awrangjeb et al. (2010b). In each rectangle, the percentage Φ of black pixels from M_g (from Fig. 3a), the average NDVI value Υ after conversion into grey-scale (from Fig. 2c) and the percentage Ψ of white pixels in the entropy mask (from Fig. 4b) are estimated. A binary flag F_b for each rectangle is also estimated, where $F_b = 1$ indicates that there are continuous black pixels in M_g along the line.

For a given line, if $\Phi < 10\%$ on both of its sides, then the line is classified as 'ground'. Otherwise, Υ and Ψ are considered for each side where $\Phi \geqslant 10\%$. If $\Upsilon > 10$ for an RGBI image or $\Upsilon > 48$ for an RGB image, and $\Psi > 30\%$ (Awrangjeb et al., 2010a) on either of the sides, then the line is classified as 'tree'. If $\Upsilon \leqslant 10$, or if $\Upsilon > 10$ but $\Psi \leqslant 30\%$, then the line is classified as 'roof ridge' if $F_b = 1$ on both sides. However, if $F_b = 1$ on one side only then it is classified as 'roof edge'. Otherwise, the line is classified as 'ground' ($F_b = 0$ on both sides), for example, for road sides with trees on the nature strip. The setup of all the parameter values has been empirically tested in Awrangjeb et al. (2010b, 2012b).

Fig. 4c and d show different classes of the extracted image lines overlaid on the orthoimage and the ground mask, respectively. For each extracted line, its two end points and slope information are

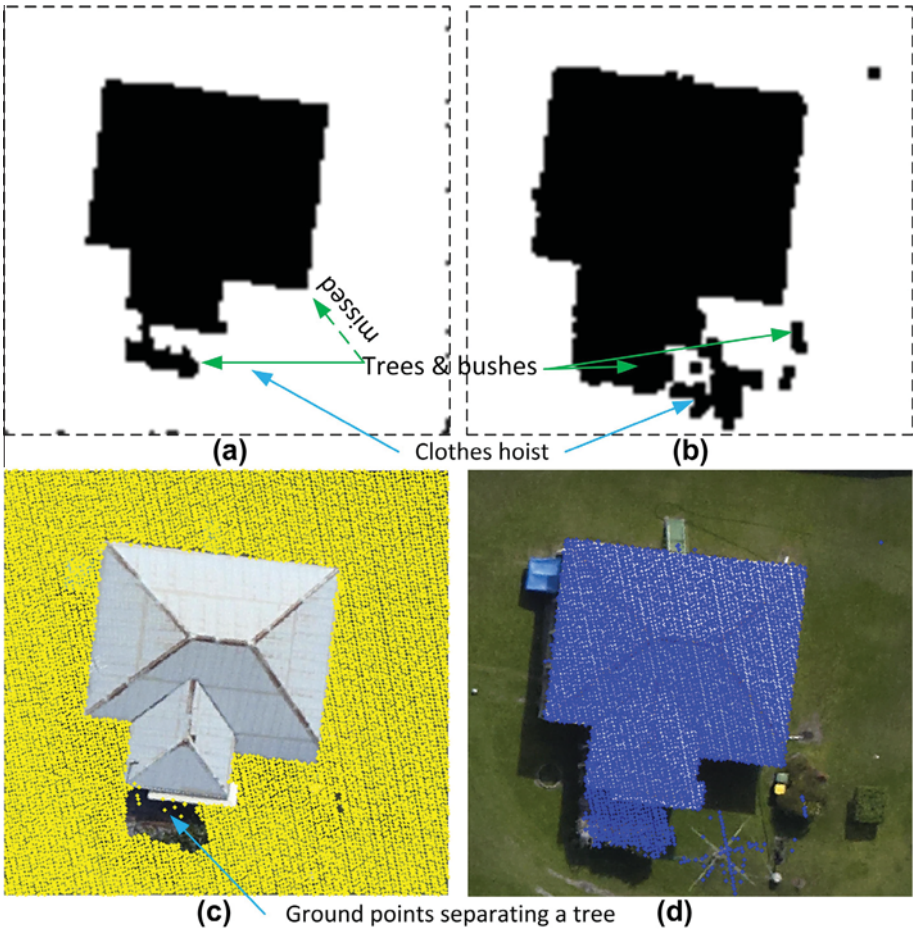

Fig. 3. For the sample scene in Fig. 2: (a) ground mask, (b) non-ground mask, (c) ground LIDAR points and (d) non-ground LIDAR points.

recorded for use in the extraction step described below. For each line, a point P_i is also recorded which indicates the side of the line where the Φ value is higher than for the other side (Awrangjeb et al., 2010b). As shown in Fig. 4c, P_i is defined with respect to the mid-point P_m of an image line when $|P_i P_m| = w_d$. For a roof edge P_i is on the corresponding roof plane and thus identifies the building side. For a roof ridge, P_i also helps in finding the other side of the line, ie the other roof plane.

3.3. Roof extraction

In this section, the roof plane extraction and false plane removal components of the extraction step in Fig. 1 are presented. In order to extract roof planes from non-ground LIDAR points, an innovative region-growing algorithm is proposed. Unlike the traditional region growing techniques which mostly require selection of appropriate seed points and are thereby sensitive to the seed points, the proposed technique depends on the extracted image lines to define robust seed regions. A seed region is defined for each roof plane with respect to a classified image line which may reside along the boundary of the plane. Logically, only the lines in classes 'roof edge' and 'roof ridge' should be considered for extraction of roof planes. Since there may be some inaccuracy in classification, lines classified as 'roof edge' or 'roof ridge' form the starting edges (baselines) for extracting roof planes, but lines in other classes may be considered if they are within the vicinity of already extracted planes and are parallel or perpendicular to a baseline.

The area, perimeter and neighbourhood of each extracted LIDAR plane, as well as the out-of-plane LIDAR spikes within its boundary, are used to decide whether it is a valid planar segment. A LIDAR plane fitted on a tree is usually small in size and there may be LIDAR spikes within its boundary. This intuitive idea is employed to remove false positive planes in the proposed procedure.

The following parameters will frequently be used throughout the extraction procedure. First, the LIDAR point spacing d_f, which is the maximum distance between two neighbouring LIDAR points, indicates the approximate LIDAR point density. Second, the flat height threshold T_f is related to the random error in the height of a LIDAR point. Ideally, two points on a truly flat plane should have the same heights, but there may be some error in their estimated LIDAR-determined heights. The parameter T_f indicates this error and it is set at 0.1 m in this study. Third, the normal distance threshold T_p to a extracted plane: if the normal distance from a LIDAR point to a plane is below $T_p = 0.15$ m this point may be included into the plane.

3.3.1. Extraction of LIDAR planes

The image lines in the 'roof edge' and 'roof ridge' classes are sorted by length. Starting from the longest line L_l in the sorted list as a baseline, the following iterative procedure is executed to extract its corresponding LIDAR plane $\top_l$, which can be located with respect to L_l's inside point P_i. When the extraction of $\top_l$ is complete, the next plane is extracted using the next longest line, and so on. If L_l is a ridge line, then the procedure is repeated to extract the plane on the other side which can be located by the mirror point of P_i with respect to L_l.

For each baseline L_l, the extraction of $\top_l$ mainly consists of following two steps: (1) estimation of plane slope and (2) iterative

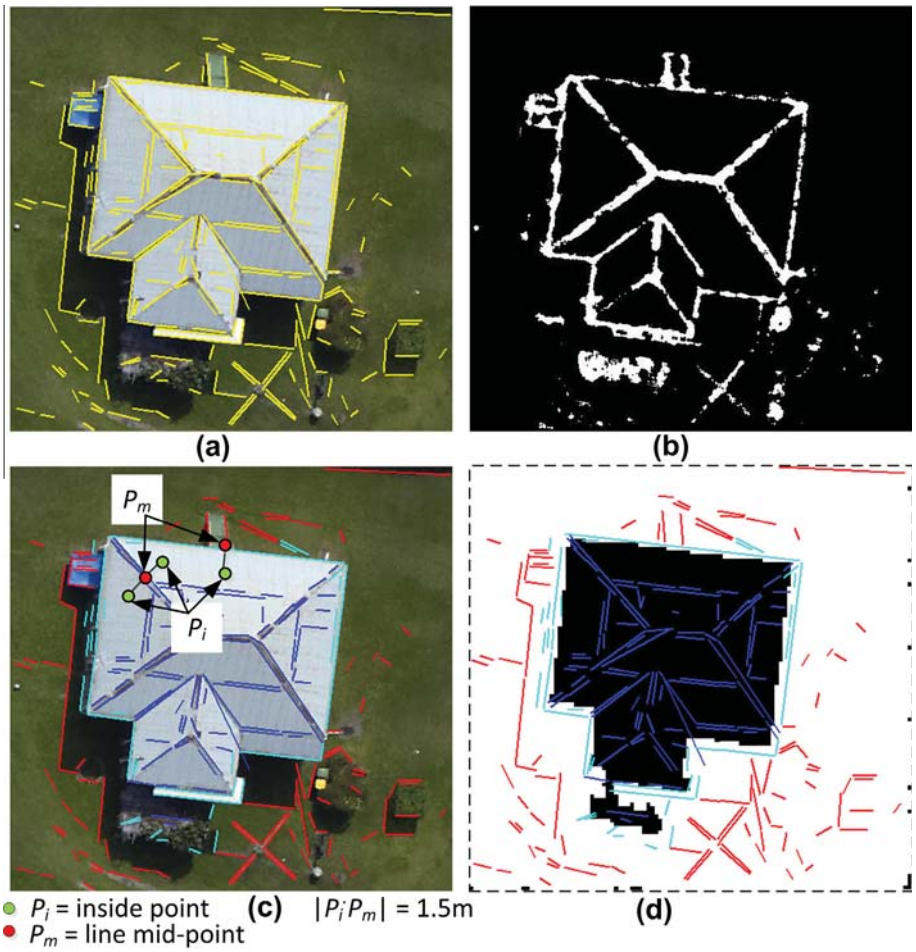

Fig. 4. Image line classification: (a) all extracted lines, (b) entropy mask generated from Fig. 2d, (c) classified lines on orthoimage and (d) on ground mask. (Classes: green: 'tree', red: 'ground', cyan: 'roof edge' and blue: 'roof ridge'.) (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

plane extraction. The slope direction θ_l of $\top_l$ with respect to L_l is first determined. Then $\top_l$ is iteratively estimated using the compatible LIDAR points near to the baseline.

(1) *Estimation of plane slope:* Fig. 5 shows some examples of different slope directions θ_l with respect to the baseline L_l. In order to estimate θ_l, all the compatible LIDAR points near to the mid-point P_m of L_l are first determined. Then the compatible points are examined to decide θ_l.
As shown in Fig. 6b, a square $P_1P_2P_3P_4$ of width w_d is considered where P_i and P_m are the mid-points of two opposite sides. The width of the square is set to w_d on the assumption that the registration error between the image and LIDAR data is at most 1 m, so that some LIDAR points inside the square may still be obtained. Let the mid-points of the other two opposite sides be P_L and P_R. Further, let S_s be the set of all LIDAR points within the square. A point $X \in S_s$ is considered compatible with $\top_l$ if it is not on any of the previously extracted planes and it has low height difference (at most T_f) with the majority of points in S_s.
Each of the four corners and four mid-points of $P_1P_2P_3P_4$ is assigned a height value which is the height of its nearest compatible LIDAR point. If P_L and P_R have similar heights (their height difference is at most T_f), but P_i and P_m do not, then there may be one of the following two cases. The slope direction θ_l is upward if the height of P_i is larger than that of P_m (Fig. 5a); otherwise, θ_l is downward (Fig. 5b). In both cases, θ_l is perpendicular to L_l. If the mean height of all the compatible points is similar to the heights of all four corners and mid-points, then θ_l is flat. As shown in Fig. 5c, there is no upward or downward slope direction on a flat plane. If P_i and P_m have similar heights, but P_L and P_R do not, then θ_l is

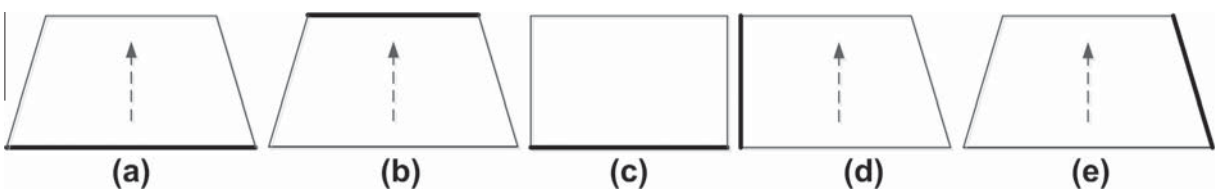

Fig. 5. Examples of different planes with respect to the baseline, where black thick line indicates the baseline and arrow indicates the slope direction with respect to the ground: (a) upward, (b) downward, (c) flat, (d) parallel and (e) undefined.

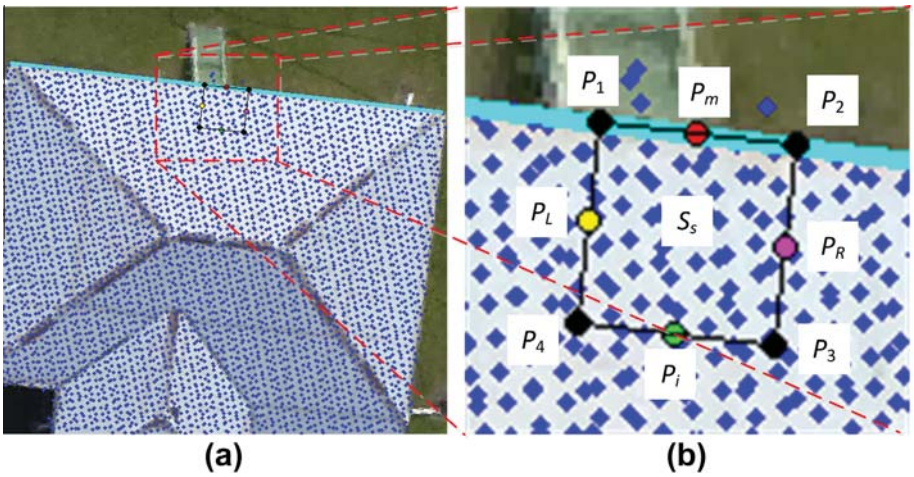

Fig. 6. Finding the slope of a plane: (a) showing baseline in cyan colour and (b) magnified version showing a square region on the plane. Raw LIDAR points within the square are examined to decide the slope of the plane.

parallel to L_l (Fig. 5d). In this case, L_l is not used to estimate a LIDAR plane, but the line P_iP_m is inserted into a list ℓ which may be used latter after all the image (edge and ridge) lines in the sorted list are used up but the corresponding plane $\top_l$ for L_l still remains unestimated. In all other cases, θ_l is undefined (see an example in Fig. 5e). In such a case, θ_l is neither perpendicular nor parallel to L_l and $\top_l$ is not a flat plane.

For the first three cases (θ_l is upward, downward or flat), the estimation of $\top_l$ is continued as described below. For the other two cases, L_l is marked as an incompatible line and the whole procedure is restarted for the next candidate line.

(2) *Iterative plane extraction:* Fig. 7 shows the image guided iterative procedure for extraction of LIDAR roof planes. Let S_p be the set of non-ground LIDAR points that have already been decided to be on $\top_l$, E_p be the set of points that are decided not to be on the current plane and U_p be the set of points which remain undecided. All three sets are initially empty. While extending the plane in four directions (shown in purple coloured arrows in Fig. 7 – Direction 1: from P_m to P_i, Direction 2: from P_i to P_m, Direction 3: towards left of P_iP_m and Direction 4: towards right of P_iP_m) S_p, U_p and E_p are updated in each iteration. Points from U_p may go any of the two other sets.

A plane is fit to the points in S_p as follows:

$$Ax + By + Cz + D = 0. \quad (1)$$

In order to obtain the four or more points used to determine the plane, a rectangle (width d_f on each side of L_l, magenta coloured solid rectangle in Fig. 7a) around the baseline is considered. If the required number of points are not found within this rectangle then the rectangle is extended iteratively, by $\frac{d_f}{2}$ each time, towards Direction 1. The points which have already been decided for any of the previously estimated planes are removed. Moreover, the points

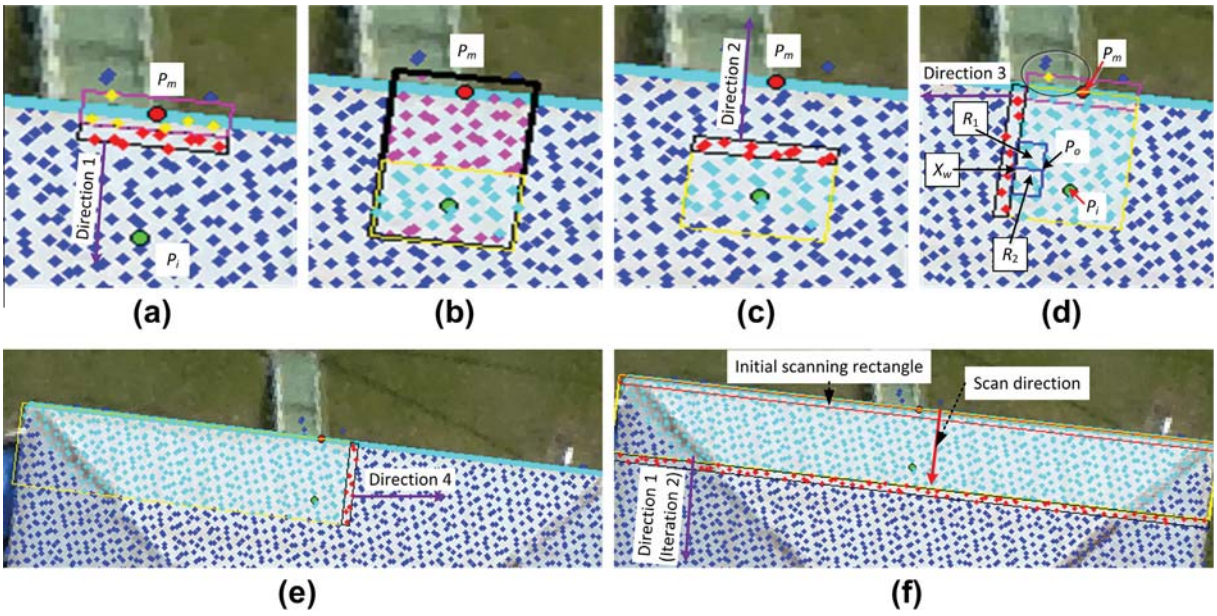

Fig. 7. Iterative extraction of a plane (coloured dots are LIDAR points, cyan coloured line is baseline, purple coloured arrows indicate directions for plane extensions): (a) initial plane around the baseline, (b) choosing the stable seed region (LIDAR points) inside plane, (c) extension of initial stable plane towards baseline, (d) extension of plane towards left, (e) extension of plane towards right and (f) plane after first iteration. In (a–f), red coloured dots inside black coloured rectangles are new candidates. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

which have high height difference (more than d_f) from the majority of points are discarded. They may constitute reflections from nearby trees or walls. Once the required number of points on the plane is obtained (yellow coloured dots in Fig. 7a), the initial plane ξ_l using Eq. (1) is constructed. These points are added to S_p. In order to check whether the new points, found during the plane extension, are compatible with θ_l, the nearest and the farthest points X_n and X_f (in S_p) from L_l are determined. For a flat plane, X_n and X_f should have similar heights. Otherwise, for a plane having an upward (downward) slope with respect to L_l, the height of X_f is larger (smaller) than that of X_n.

The plane is then iteratively extended towards Direction 1 by considering a rectangle of width d_f outside the previous rectangle. All the LIDAR points (see red coloured points inside the black coloured rectangle in Fig. 7a) within the currently extended rectangle are the candidate points and sorted according to their distance from L_l. Starting from the nearest point X_w in the sorted list, the following conditions are executed sequentially for one point at a time.

- *Unused*: X_w should not be used for estimating any of the previously extracted planes.
- *Plane compatible*: The plane compatibility condition can be tested in one of following three ways. First, the mean height of all the neighbouring points in S_p is similar to the height of X_w. A circular neighbourhood of radius $3d_f$ is considered. Second, the estimated height of X_w using Eq. (1) is similar to its LIDAR height. Third, its normal distance to the plane is at most T_p.
- *Slope compatible*: For a flat plane, X_w has a height similar to both X_n and X_f. Otherwise, for a plane having an upward (downward) slope with respect to L_l, the height of X_w is larger (smaller) than that of X_f.

Candidate points which were decided to be on any previous planes are first removed from the candidate point set. From the rest of the points, points which do not satisfy the plane compatibility condition are included into E_p, points which do not satisfy the slope compatibility condition are included into U_p and points which satisfy all three conditions are included into S_p.

After deciding all the points in the extended rectangle (black rectangle in 7a), the nearest and furthest points X_n and X_f are updated and the plane extension is continued towards Direction 1. In the first iteration, when the width w_e of the extended plane exceeds $w_i + 2d_f$, where $w_i = 1$ m, the extension in Direction 1 is stopped and only points having distances more than w_i, but less than $w_i + 2d_f$, from L_l are kept in S_p (also X_n and X_f are updated) and others are discarded. A stable seed region is thus obtained for $\top_l$. For example, the cyan and magenta coloured points inside the black coloured rectangle in Fig. 7b form the initial plane. The plane width at this moment is $w_e = 2.16$ m. So, only the cyan coloured points are kept in S_p and magenta coloured points are discarded. For planes having small w_e in the first iteration this check does not succeed, so they are continued with all the points in S_p found so far.

Then the plane is grown towards Direction 2 (Fig. 7c). The red coloured points inside the black rectangle (width d_f) of Fig. 7c are new candidates. The cyan coloured dots within the yellow rectangle in Fig. 7d are decided to be on the plane at the end of the extension towards Direction 2. The yellow coloured point within the black coloured circle is now found to be compatible with the plane, but as shown in Fig. 7a, this point was initially selected to be on the plane.

Points in U_p are now decided. These points are sorted according to their distance from L_l. Starting from the nearest point, if a point $X_u \in U_p$ satisfies all three conditions above, it is included into S_p. If X_u fails the plane compatibility test, it is decided to be on another plane and therefore included in E_p. Otherwise, X_u still remains undecided.

Thereafter the plane is extended towards Direction 3 (see Fig. 7d). The red coloured points inside the black rectangle (width d_f) in Fig. 7d are the new candidates, which are now sorted according to their distances from P_iP_m. Starting from the nearest point X_w (shown in Fig. 7d) in the sorted list, all three conditions are executed for one point at a time. The slope compatibility test is modified as follows. A rectangular neighbourhood is considered at the right side of X_w where there are points which have already been decided to be on the plane. As shown in Fig. 7d, a point P_o inside the already extracted plane rectangle (yellow coloured rectangle) is obtained, where X_wP_o is parallel to L_l and $|X_wP_o| = 2d_f$. Two rectangles R_1 and R_2 are formed below and above X_wP_o. Let S_1 and S_2 be the sets of LIDAR points from S_p which reside inside R_1 and R_2 respectively. For a flat plane, the height of X_w should be similar to the mean height of $S_1 \cup S_2$. For a plane having an upward (downward) slope, the height of X_w should be higher (smaller) than the mean height of S_1 but smaller (higher) than the mean height of S_2. Fig. 7e shows the extended plane (cyan coloured dots inside the yellow coloured rectangle) when extension towards Direction 3 ends.

The plane is then extended towards Direction 4 using the same conditions discussed above. The red dots within the black rectangle in Fig. 7e are the new candidates. Fig. 7f shows the extended plane (cyan coloured dots inside the yellow coloured rectangle) when extension towards Direction 4 ends. Points in U_p are again checked to decide (following the same conditions discussed before) and the plane equation ξ_l is updated.

After extending the plane in all four directions, all the LIDAR points, which are neither in S_p nor in E_p, within the plane rectangle (yellow coloured rectangle in Fig. 7f) are re-examined to ascertain whether they can still be added to the current plane. In order to find these candidates, a rectangular window of width d_f (red coloured rectangle) is scanned from one end of the current plane rectangle (yellow coloured rectangle) to the other end (see Fig. 7f). For these candidates, the *plane compatibility* condition is modified as follows. First, candidates whose normal distances to ξ_l are at most T_p are chosen. Second, a recursive procedure is employed where a candidate point is decided to lie in the plane if its closest neighbour has already been decided to lie in the plane. A maximum radius of $2d_f$ is allowed for a circular neighbourhood. Finally, points in the undecided set U_p are again tested to decide. Fig. 8 shows an example for a plane in the sample data set where the red dots within the highlighted rectangle in Fig. 8a (45 red points) were re-examined in the first iteration of the extraction procedure. All of these points were found compatible and thus included in the plane, as shown in Fig. 8b.

At this point, one iteration of the iterative plane extraction procedure is over and the next iteration starts with the new candidates shown as the red dots within the black coloured rectangle in Fig. 7f. Subsequently, the plane is extended again to all four directions. If no side is extended in an iteration, the iterative procedure stops. For example, the extraction of the current plane of the sample data set is completed in the second iteration, as shown in Fig. 9a.

An extracted plane must have a minimum number of LIDAR points and a minimum width in order to be considered as a roof plane. This test removes some of the false positive planes on trees. The required minimum number of points is $n_m = \left(\frac{W_h}{d_f}\right)^2$, where $W_h = 1.5$ m (half of minimum building width W_m) and the minimum plane width is $l_m = 1$ m.

All the image lines which reside within an extended plane rectangle (black coloured rectangle in Fig. 9a, whose width and height of the plane rectangle are increased by 1 m on each side) and are

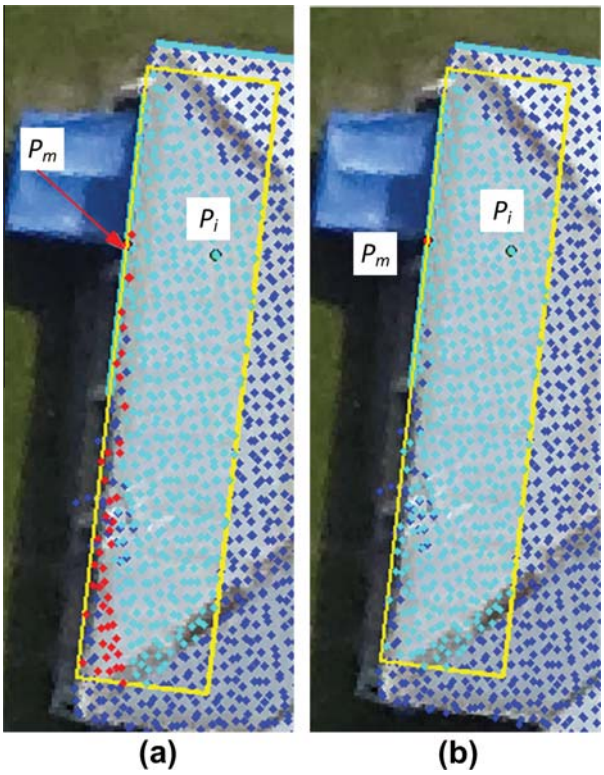

Fig. 8. Scanning unused LIDAR points within a plane rectangle (coloured dots are LIDAR points and cyan coloured line is baseline): (a) Plane at the end of iteration 1 (cyan coloured dots inside yellow coloured plane rectangle form current plane, red coloured dots indicate unused candidates) and (b) all the unused candidates are now assigned to the plane after they are checked for compatibility.

either parallel or perpendicular to L_l are kept in a queue Q for priority processing. Fig. 9a shows 21 such lines in yellow. Lines in Q are processed starting first with the longest line. Consequently, the extraction of a neighbouring plane is started with a new baseline shown as a thick yellow line in Fig. 9a. If Q is empty then the next longest roof edge or ridge line is used as a baseline to extract a new plane. Finally, when all the roof edge and ridges lines are decided, new lines in $\ell$are processed if there is any true planes left unconstructed. Fig. 9b shows all the extracted planes for the sample data of Fig. 2.

3.3.2. Removal of false planes

In order to remove false positive planes, mostly constructed on trees, a new rule-based procedure is proposed. For an extracted LIDAR plane, its area, perimeter and neighbourhood information, as well as any LIDAR spikes within its boundary are used to decide whether it is a false alarm. A LIDAR plane fitted on a tree is usually small in size and there may be some LIDAR spikes within its boundary.

In order to estimate the above properties, it is important to obtain the boundary of each of the extracted planes. For a given set of LIDAR points S_p of an extracted plane $\top_l$, a binary mask M_b is formed. The boundary of the plane is the Canny edge around the black shape in M_b. Fig. 10a shows the generated mask and the boundary Δ_l for the first plane of the sample data set, and Fig. 10b shows all the boundaries for the sample data set.

The area and perimeter of $\top_l$, determined via boundary Δ_l, can thus be estimated. The perimeter is simply the sum of consecutive point-to-point distances in Δ_l, and to obtain its area, a Delaunay triangulation is formed among the points in Δ_l and the triangles that reside outside the boundary are removed (red colour triangles in Fig. 10c). Therefore, the area of the plane is the sum of areas of the remaining triangles (blue coloured in Fig. 10c).

Thereafter, it is necessary to find the neighbouring planes for a given plane and to group these neighbouring planes. A group of neighbouring planes represent a complete building or a part of a building. If the planimetric boundary Δ_1 of $\top_1$ passes within a distance of $2d_f$ from Δ_l, then $\top_l$ and $\top_1$ are initially considered neighbouring planes. Three 3D lines are then estimated for the two neighbours: l_l for neigbouring LIDAR points in Δ_l, l_1 for neigbouring LIDAR points in Δ_1, and the intersection l_i of the two planes (using equations of planes ξ_l and ξ_1 according to Eq. (1)). These two planes are real neighbours if l_i is parallel to both l_l and l_1 and the perpendicular distances between the parallel lines should not exceed 1 m. For example, Planes A and B in Fig. 11 are two neighbours and three lines l_A, l_{B1} and l_{AB} are estimated, where l_A is estimated using the neighbouring LIDAR points from Plane A, l_{B1} from Plane B, and l_{AB} is the intersection line found by solving two plane equations. It can be seen that l_{AB} is parallel to both l_A and l_{B1} and it is close to them. Consequently, Planes A and B are found to be real neighbours. Another example, Planes B and C are initially found to be neighbours and their intersection line l_{BC} is parallel to both l_{B2} and l_C. However, since l_{BC} is far from both l_{B2} and l_C, Planes B and C are decided not to be real neighbours.

By using the above analysis all of the extracted planes for the sample data set are formed into two groups, as shown in

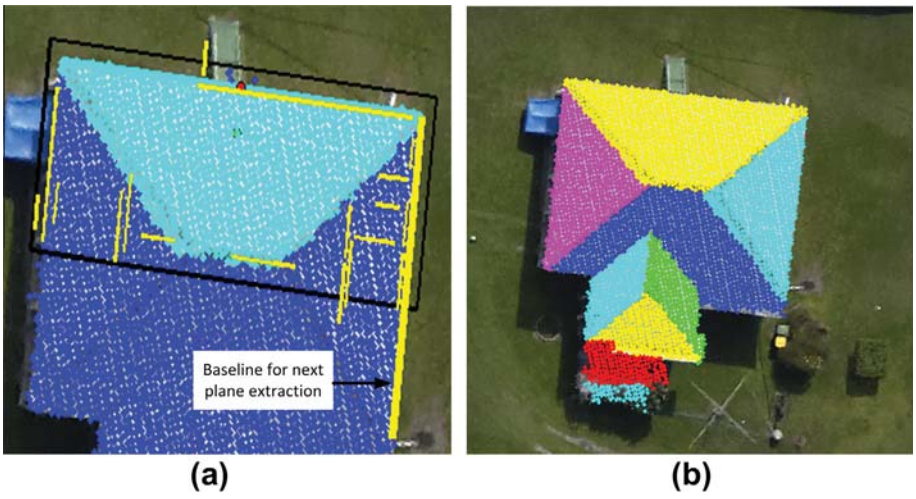

Fig. 9. Results of iterative plane extraction: (a) first plane and (b) all planes.

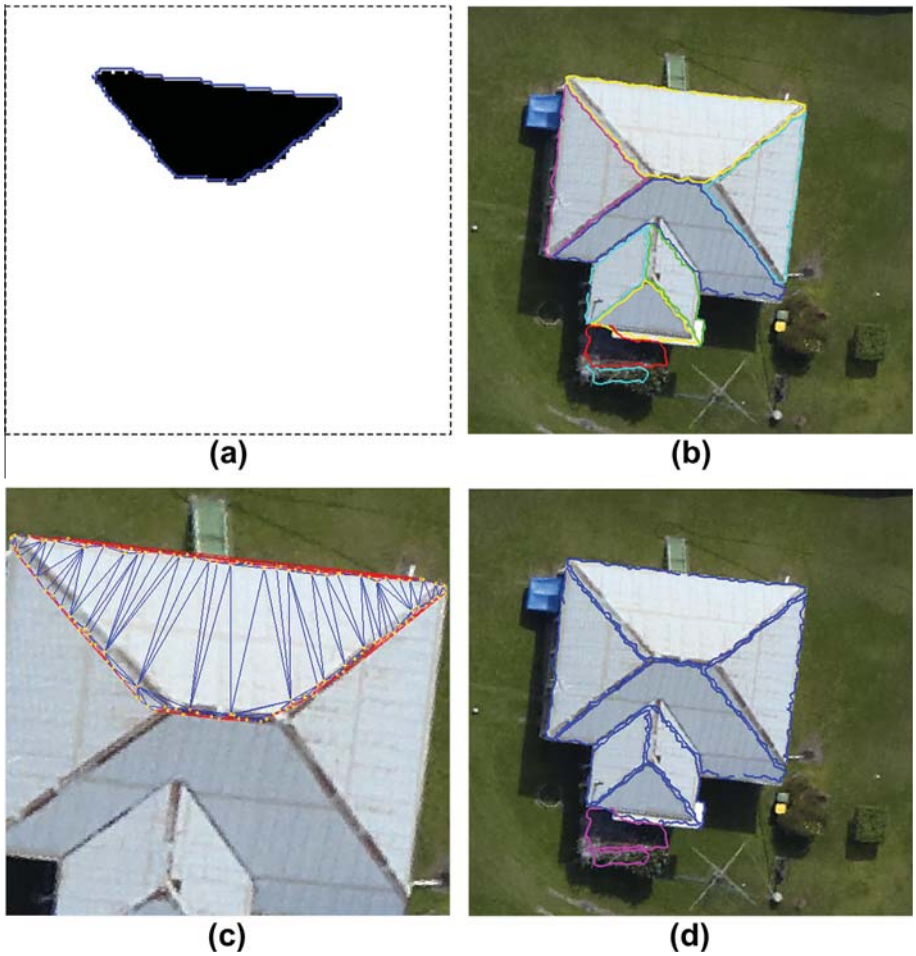

Fig. 10. Plane boundaries: (a) extraction of boundary around a plane's binary mask, (b) all plane boundaries for planes in Fig. 10b, (c) Delaunay triangulations of plane's boundary points to estimate plane area and (d) Groups of neighbouring planes, all blue boundaries are in one group and all magenta in other group. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

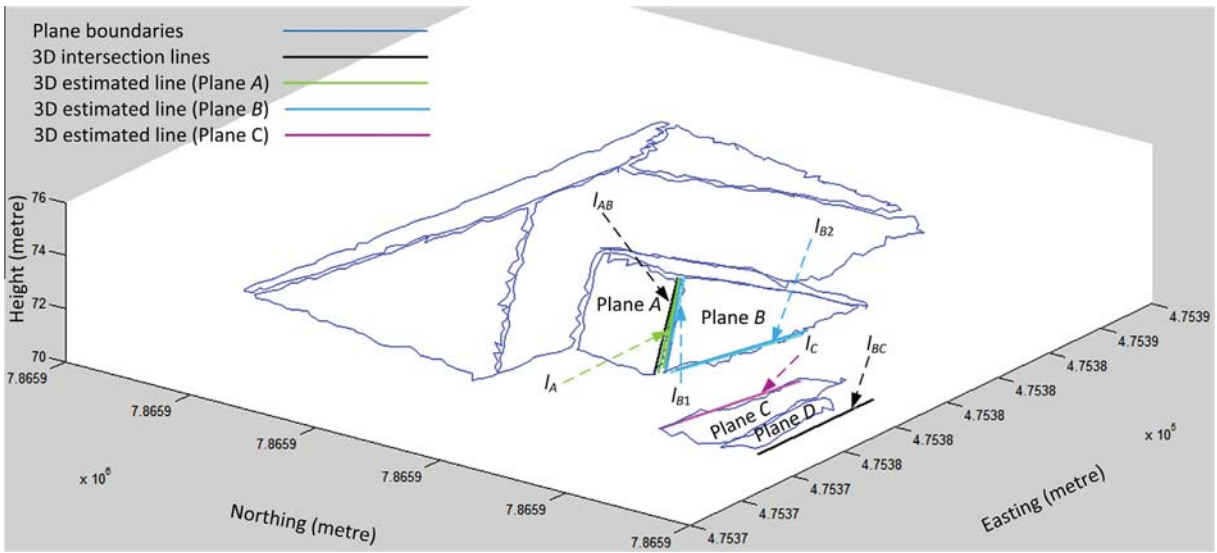

Fig. 11. Finding the real neighbouring planes of a given plane.

Fig. 10d, one being shown in blue colour and the other in magenta colour.

In order to remove the false positive planes, the following tests are executed sequentially for each extracted plane $\top_l$.

- *Point and height tests*: A 3D cuboid is considered around $\top_l$ 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 within the cube and their height values are estimated using the plane equation ξ_l. If at least one estimated height z_e is too low ($z_m - z_e > T_z$) or too high ($z_e - z_M > T_z$), then $\top_l$ is decided as a false plane.

- *With no neighbours test*: If $\top_l$ does not have any neighbouring planes, it must be at the size of the required minimum building width W_m. If its area is less than 9 m^2 or its perimeter is less than 12 m, then it is marked as a false plane.
- *With neighbours test*: After the above test all the true positive planes smaller than 9 m^2 have at least one neighbouring plane. It is assumed that an individual $\top_l$, when it has neighbouring planes on the same roof, should have at least 1 m^2 area in order to be extracted properly. Therefore, if $\top_l$ is less than 1 m^2, it is removed as a false plane.
- *Group test – area*: If at least one plane in a neighbourhood group has an area of at least 9 m^2, all planes in the group are accepted as true planes.
- *Group test – lines*: Through consideration of a rectangular neighbourhood around a group of planes, image lines within the neighbourhood are identified. If there is at least a line of more than $1.5W_m$ or if there are at least a pair of lines (with length of at least W_m) which are parallel or perpendicular to each other, then all planes in the group are considered to be true planes.
- *Group test – no-ground points*: If a plane $\top_l$ contains no-ground points continued from a neighbouring plane that has already been decided as a true plane, then $\top_l$ is designated a true plane as well. Continuous non-ground points are indicated by the continuous black pixels in the ground mask. In a recursive procedure, a small plane of a group can be designated a true plane if its neighbouring plane in the group is also found to be as a true plane. Consequently, this test allows small roof planes, for example, pergola which are physically connected to the main building but may not have real plane intersections with the main building, to be treated as parts of a complete building roof.

  Each group of planes needs to satisfy any of the above group tests. All other groups at this point are decided to be non-complaint and the planes in those groups are considered to be false.
- *Height test*: Finally, all surviving planes which are smaller than 3 m^2 in area are subject to a further test to remove nearby planes on trees. For such a plane, if the difference between maximum and minimum boundary heights is more than 1 m then this plane is removed as a false plane. This test removes randomly oriented planes on nearby trees. Because these tree planes are in the group containing true roof planes, they may survive after all of the above group tests.

During the application of the above removal procedure, the first group of planes (shown in blue colour in Fig. 10d) satisfied the area test since it had at least 1 plane whose area was more than 9 m^2. However, the second group of planes (shown in magenta colour in Fig. 10d) did not satisfy the area test as the total area of its planes was less than 9 m^2. Nevertheless, since the second group had continuous non-ground points with the first group (see Fig. 4d), the second group was also designated part of the complete building roof.

4. Performance study

In the performance study conducted to assess the proposed approach, two data sets from two different areas were employed. The objective evaluation followed a previously proposed automatic and threshold-free evaluation system (Awrangjeb et al., 2010b,c).

4.1. Data sets

The test data sets cover two urban areas in Queensland, Australia: Aitkenvale (AV) and Hervey Bay (HB). The AV data set comprises two scenes. The first scene (AV1) covers an area of 108 m × 80 m and contains 58 buildings comprising 204 roof planes. The second (AV2) covers an area of 66 m × 52 m and contains five buildings comprising 25 roof planes. The HB data set has one scene and covers 108 m × 104 m and contains 25 buildings consisting of 152 roof planes. All three data sets 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.

LIDAR coverage of AV comprises first-pulse returns with a point density of 35 points/m^2, a spacing of 0.17 m in both in- and cross-flight directions. For HB, the first-pulse LIDAR points have a point spacing of 0.17 m in-flight and 0.47 m in cross-flight directions (13 points/m^2). The AV and HB image data comprise RGB colour orthoimagery with resolutions of 0.05 m and 0.2 m, respectively. Bare-earth DEMs of 1 m horizontal resolution cover both areas. For the data sets having only RGB color orthoimagery the pseudo-NDVI image instead of the NDVI image was employed, following the process in Rottensteiner et al. (2005).

In order to empirically test the sensitivity of the algorithmic parameters, the AV2 scene has been employed. While the original LIDAR point density of this data set is 35 points/m^2, it has been resampled to different densities for tests conducted: 16, 11, 8, 6 and 4 points/m^2.

Two dimensional reference data sets were created by monoscopic image measurement using the Barista software (Barista, 2011). All visible roof planes were digitized as polygons irrespective of their size. The reference data included garden sheds, garages, etc. These were sometimes as small as 1 m^2 in area.

As no reference height information was available for the building roof planes, only the planimetric accuracy has been evaluated in this study. Since each of the extracted plane boundaries consists of raw LIDAR points, it can be safely assumed that the accuracy in height depends completely on the input LIDAR data.

Table 1
Parameters used by the proposed roof extraction method.

Parameters	Values	Sources
Ground height H_g	DEM height	input LIDAR
Height threshold T_h	H_g + 2.5 m	Rottensteiner et al. (2004)
Min. building width W_m	3 m	Awrangjeb et al. (2010b)
Rect. neighbourhood width w_d	$\frac{W_m}{2}$	Awrangjeb et al. (2010b)
Min. image line length l_m	$\frac{W_m}{3}$	related to W_m
Line classification thresholds	(see Section 3.2.2)	Awrangjeb et al. (2012b)
LIDAR point spacing d_f	(from input LIDAR data)	input LIDAR data
Flat height threshold T_f	0.10 m	this paper
Normal distance threshold T_p	0.15 m	this paper
Width of extending rectangle	d_f	related to d_f
All LIDAR neighbourhood sizes	proportional to d_f	related to d_f
Min. points on a plane	$\left(\frac{W_m}{2d_f}\right)^2$	related to W_m, d_f
Min. building area	9 m^2	related to W_m
Min. building perimeter	12 m	related to W_m
Min. plane area	1 m^2	related to l_m
Spike height threshold T_z	1.5 m	this paper

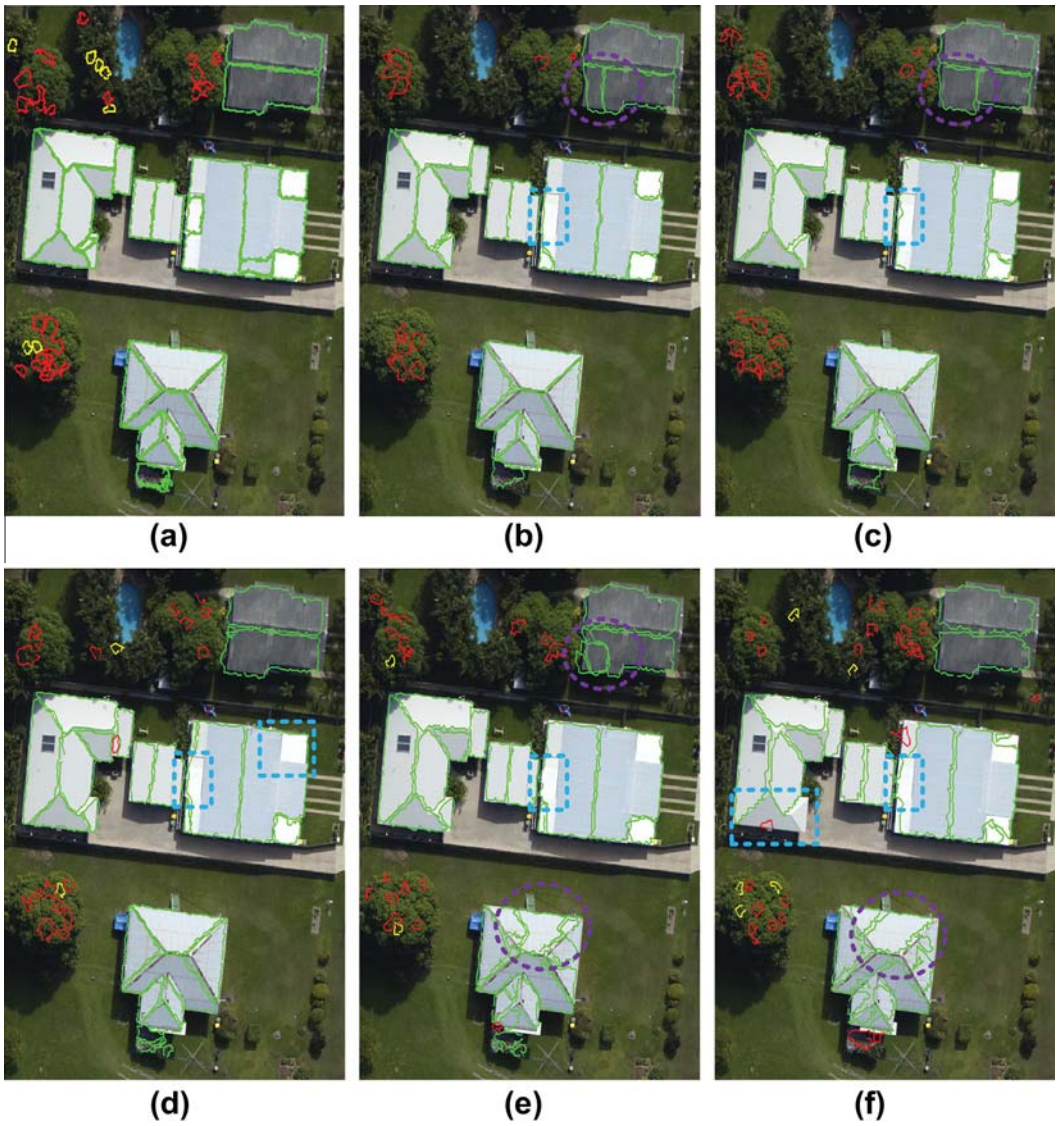

Fig. 12. Roof plane extraction results for the AV2 data set under different LIDAR point density: (a) original density, 35 points/m^2 (LIDAR point spacing d_f = 0.17 m), (b) 16 points/m^2 (d_f = 0.25 m), (c) 11 points/m^2 (d_f = 0.30 m), (d) 8 points/m^2 (d_f = 0.35 m), (e) 6 points/m^2 (d_f = 0.40 m) and (f) 4 points/m^2 (d_f = 0.50 m). Green: extracted roof planes, other colours: false planes mostly extracted on trees and therefore removed.

Table 2
Object-based evaluation results for the AV2 data set in percentages under different LIDAR point density in points/m^2 (C_m = completeness, C_r = correctness, Q_l = quality, C_{rd} = detection cross-lap rate and C_{rr} = reference cross-lap rate).

LIDAR density	C_m	C_r	Q_l	C_{rd}	C_{rr}
35	100	96.2	96.2	0.0	8
16	100	100	100	3.8	8
11	100	100	100	7.7	12
8	100	100	100	11.1	12
6	100	100	100	8.8	28
4	95.2	100	95.2	12	20
Average	99.2	99.4	98.6	7.2	14.7

Table 3
Pixel-based evaluation results for the AV2 data set in percentages under different LIDAR point density in points/m^2 (C_{mp} = completeness, C_{rp} = correctness, Q_{lp} = quality, B_f = branching factor and M_f = miss factor).

LIDAR density	C_{mp}	C_{rp}	Q_{lp}	B_f	M_f
35	96.6	96.9	93.7	3.2	3.6
16	95.6	97.7	93.5	2.3	4.6
11	95.9	97.5	93.6	2.5	4.4
8	96.2	96.9	93.4	3.2	3.9
6	94.9	97.2	92.4	2.9	5.4
4	88.4	97.7	86.5	2.4	13.2
Average	94.6	97.3	92.2	2.8	5.9

4.2. Evaluation metrics

The planimetric accuracy has been measured in both object and image space. In object-based evaluation the number of planes has been considered – whether a given plane in the reference set is present in the detection set. Five indices are used for object-based evaluation. *Completeness* C_m, also known as *detection rate* (Song and Haithcoat, 2005) or *producer's accuracy* (Foody, 2002), *correctness* C_r, also known as *user's accuracy* Foody (2002) and *quality* Q_l have been adopted from Rutzinger et al. (2009). *Detection cross-lap rate* is defined as the percentage of detected planes which overlap more than one reference planes. *Reference cross-lap rate* is defined as the

percentage of reference planes which are overlapped by more than one detected plane (see Awrangjeb et al. (2010c) for formal definitions).

In pixel-based image space evaluation the number of pixels has been considered – whether a pixel in a reference plane is present in any of the detection planes. A total of 5 pixel-based evaluation indices are used, these being: *completeness* C_{mp}, also known as *matched overlay* (Song and Haithcoat, 2005) and *detection rate* (Lee et al., 2003), *correctness* C_{rp} and *quality* Q_{lp} from Rutzinger et al. (2009); and *branching factor* B_f and *miss factor* M_f from Lee et al. (2003).

4.3. Parameter sensitivity

Table 1 shows all the parameters used by the proposed roof extraction method. Many of the parameters have been directly adopted from the existing literature. Some of them, e.g. LIDAR point spacing, are related to the input data. Many of the parameters are dependent on other parameters, e.g. minimum building area 9 m^2 is related to the minimum building width $W_m = 3$ m. The spike height threshold T_z is a new parameter and is used during the point test in Section 3.3.2 to remove randomly oriented planes on trees. It is an independent parameter. The other two parameters, the flat height threshold T_f and the plane normal distance threshold T_p, are used while extracting the planar roof segments. Both T_f and T_p are related to the input LIDAR data where there is always some accuracy limitation as to why two LIDAR points on a flat plane may have different height values.

While T_f is somewhat new and directly applied to the heights of the neighbouring LIDAR points to a given LIDAR point, T_p has been used in the literature frequently and is applied to the perpendicular distances from the LIDAR points to the estimated planes. Sampath and Shan (2010) showed that the perpendicular distances from the segmented LIDAR points to the plane was in the range of 0.09–0.60 m, Chen et al. (2006) found this error to be in the range of 0.06 m to 0.33 m and Habib et al. (2010) set a buffer of 0.4 m (twice the LIDAR accuracy bound) on both sides of the plane to obtain the coplanar points.

Three sets of different values (in metres) were tested while setting the values for the three parameters T_f, T_p and T_z. For both T_f and T_p the test values were 0.05, 0.1, 0.15, 0.2, 0.25 and 0.3 m. For T_z the test values were 0.2, 0.4, 0.5, 1.0, 1.5, 2.0, 2.5, 3.0 and 3.5 m. While for T_f and T_p the object- and pixel-based qualities were considered, for T_z the number of planes was employed since the quality values were close at different test values. It was observed that for both T_f and T_p the object-based qualities were similar but the pixel-based qualities changed slightly at different test values. The maximum pixel-based qualities were observed at $T_f = 0.1$ m and $T_p = 0.15$ m. At low test values for T_z a large number of trees (small to large in area) were removed. In addition, a small number of roof planes were also removed. However, at high test values only the trees (large in area) were removed but no tree planes were removed. At $T_z = 1.5$ m the number of removed trees (medium to large in area) was moderate and no roof planes were removed. The trees which are small in area were removed using other rules discussed in Section 3.3.2. As a result, the chosen values are $T_f = 0.1$ m, $T_p = 0.15$ m and $T_z = 1.5$ m.

In order to test how the algorithm performs when the parameters (eg, neighbourhood sizes and the minimum number of points on a roof plane, as shown in Table 1) related to the input LIDAR point spacing d_f change, the LIDAR density was progressively decreased in the AV2 scene to as low as 4 points/m^2.

Fig. 12 shows the extraction results under different LIDAR densities in the AV2 scene. Tables 2 and 3 show the objective evaluation results in object- and pixel-based metrics respectively. The object-based completeness, correctness and quality were more than 95% in all LIDAR point densities, however there were crosslaps as shown in Table 2.

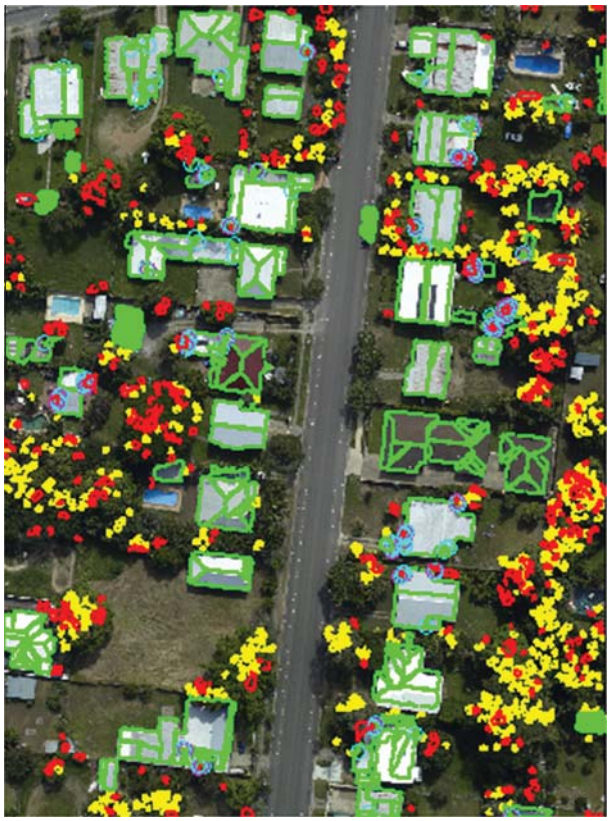

Fig. 13. Results for the AV1 data set. Green: extracted roof planes, other colours: false planes mostly extracted on trees and therefore removed. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Fig. 14. Results for the HB data set. Green: extracted roof planes, other colours: false planes mostly extracted on trees and therefore removed.

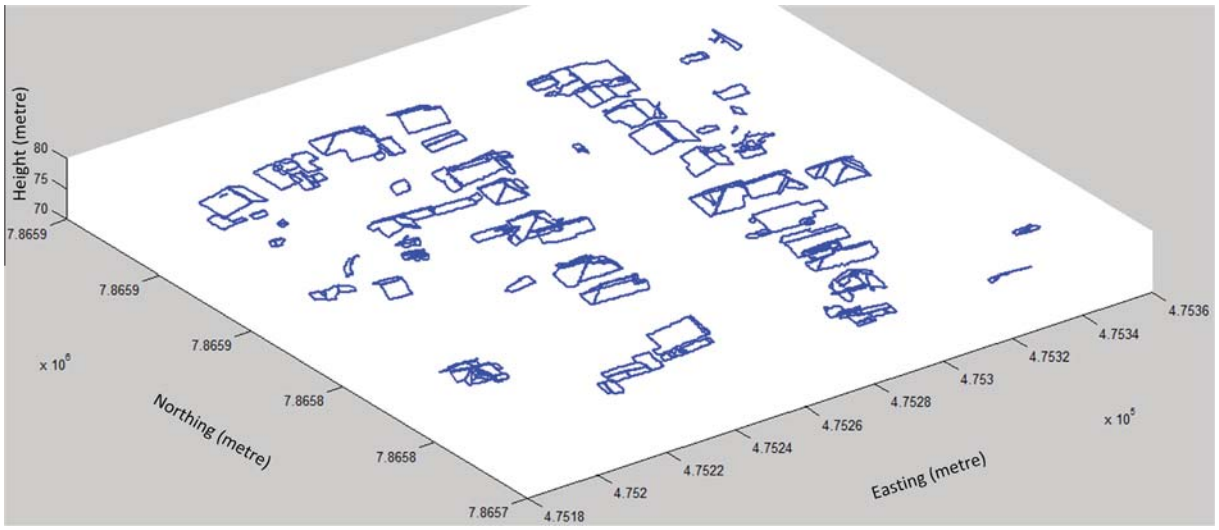

Fig. 15. Three-dimensional building models from the AV1 data set.

Overall, it has been observed that when the density of the LIDAR data decreases, many of the planes are extracted. However, some of the larger planes have been extracted into more than one small component (over-segmentation) and some of the small planes have not been extracted as separate planes (under-segmentation), which in turn causes reference and detection cross-laps. Fig. 12 shows that each of the planes inside the dashed purple coloured circles has been extracted as more than one detection planes. This has moderately increased the reference cross-lap rates, specially when the LIDAR density is low (see Table 2). Moreover, some of the small planes, shown in cyan coloured dashed rectangles in Fig. 12, have been merged with neighbouring large planes, which in turn gradually increased the detection cross-lap rates for low LIDAR density cases (see Table 2).

Thus from the results in Tables 2 and 3, it can be concluded that the proposed extraction algorithm works well when the LIDAR point density decreases gradually, but in low LIDAR density cases its performance deteriorates moderately. Consequently, the parameters used in this paper were found to be somewhat sensitive in the experimentation conducted.

4.4. Results and discussion

Figs. 13 and 14 show the extracted planes for the AV and HB data sets and Figs. 15 and 16 show the corresponding extracted 3D building models.

It can be observed that almost all the roof planes are correctly extracted and the false planes on trees are correctly removed. However, due to nearby trees or small dormers on the roof, there are cases where over-segmentation causes some of the true planes to be extracted in two or more small components. Figs. 17 and 18 illustrate some of these cases: small complex structures on roof tops have been correctly extracted (Fig. 17a); occluded and shaded small to medium sized roofs are extracted correctly (Fig. 17b); small planes extracted on trees have been successfully removed (Fig. 17b); roofs smaller than 9 m^2 have been removed (Fig. 17c and d); low height roofs are missed (Fig. 17c); complex building neighbourhood structure has been correctly extracted; planes as small as 1 m^2 that reside among big planes are correctly extracted (Fig. 18a and b); close but parallel neighbouring planes have been separated and correctly extracted (Fig. 18b); and finally, the

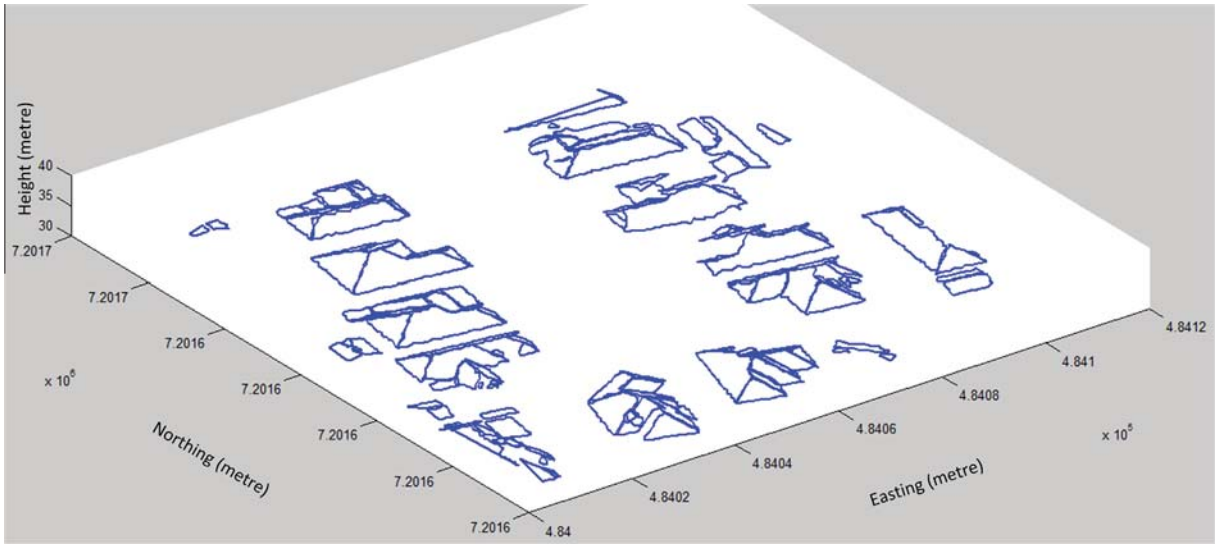

Fig. 16. Three-dimensional building models from the HB data set.

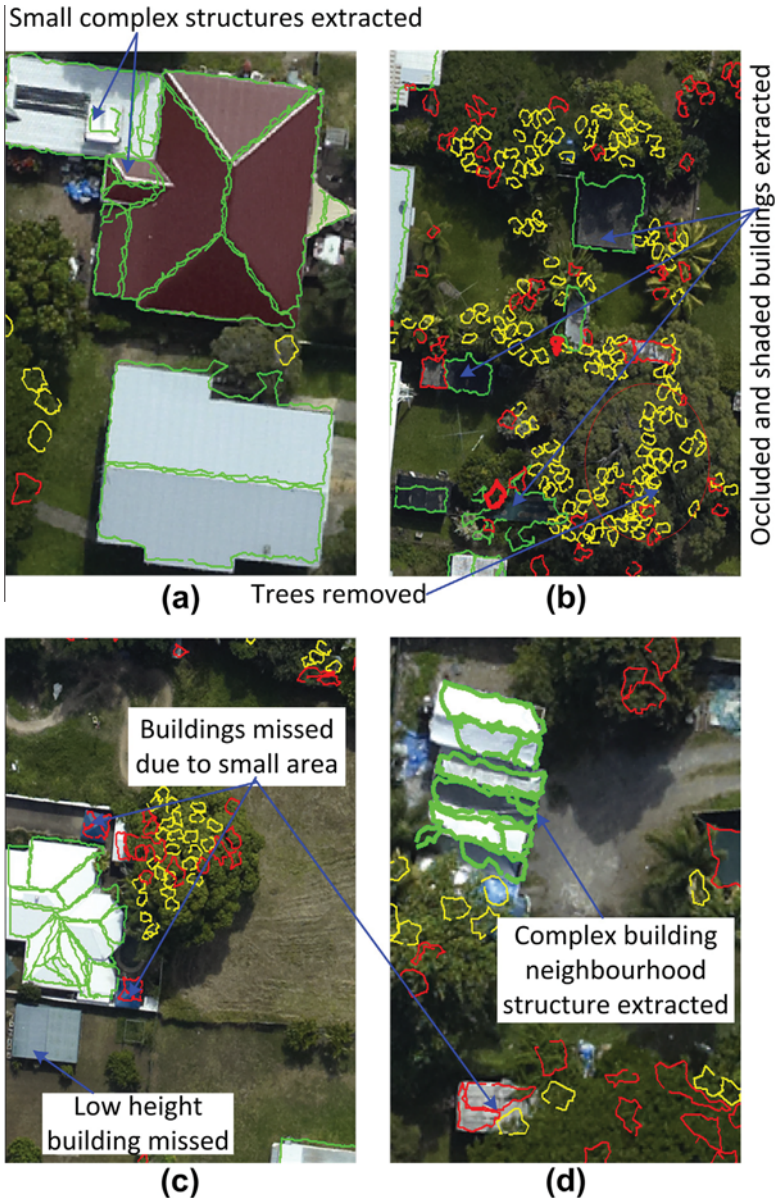

Fig. 17. Some special cases of roof extraction in the AV1 data set. Green: extracted roof planes, other colours: false planes mostly extracted on trees and therefore removed. Purple coloured dotted circles: more than one detection planes for a given reference plane, cyan coloured dotted rectangle: plane has not been extracted separately. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

proposed algorithm performs well in spite of there being registration errors of 1–2 m as shown in Fig. 18b between the LIDAR data and orthoimage.

Tables 4 and 5 show the objective evaluation results using object- and pixel-based metrics, respectively. The proposed algorithm offers slightly better performance on the AV data set than on the HB data set in terms of both object- and pixel-based completeness, correctness and quality. It shows higher detection cross-lap rate in the HB data set, but higher reference cross-lap rate in the AV data set. These phenomena can be explained as follows. In the AV data set, there are a few cases where neighbouring trees partially occluded some of the planes. As a result, more than one plane has been extracted on each of these occluded planes and the reference cross-lap rate has been increased. In the HB data set, there are many small planes on complex roof structures and some of them have been merged with the neighbouring larger planes. Thus the detection cross-lap rate has been increased for the HB data set.

In the HB data set, the branching and miss factors are high. The large registration error between the LIDAR and orthoimage increased both of these error rates. In addition, some small planes were missed, which has increased the miss factor as well.

Note that in both data sets (Figs. 13 and 14) there were many overgrown and undergrown regions. Nevertheless, the object-based completeness and correctness were more than 98%, as shown in Tables 4 and 5. This is attributable to the threshold-free evaluation system (Awrangjeb et al., 2010b,c), which determines a true detection based on the largest overlap between a detection and a reference entity. Then the over- and under-segmentation cases are explicitly expressed by indices such as reference and detection cross-lap rates in the object-based evaluation (Table 4), and branching and miss factors in the pixel-based evaluation (Table 5). Consequently, although the proposed roof extraction technique showed high correctness and completeness, the high reference cross-lap rate in the AV data set indicated that there

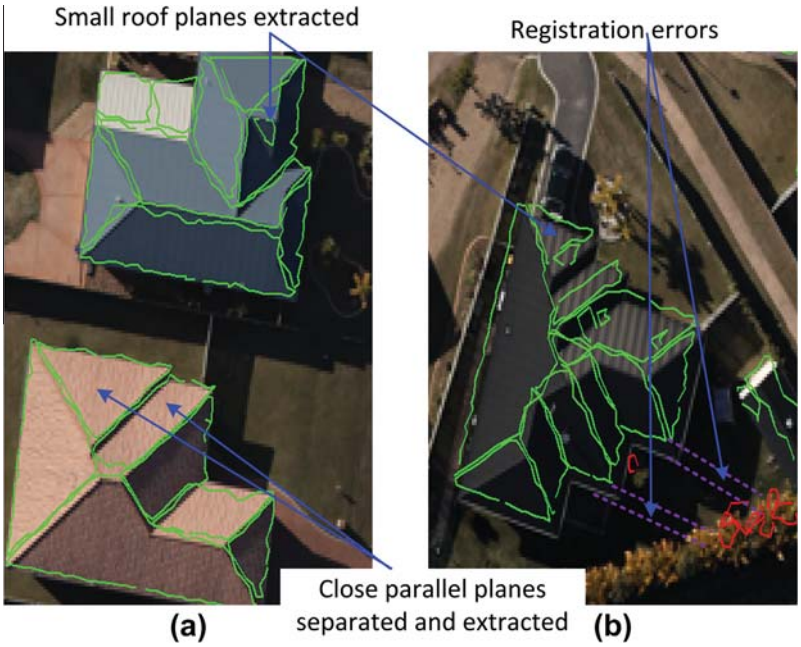

Fig. 18. Some special cases of roof extraction in the HB data set. Green: extracted roof planes, other colours: false planes mostly extracted on trees and therefore removed. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Table 4
Object-based evaluation results in percentages for two test data sets: AV1 – first Aitkenvale scene, AV2 – second Aitkenvale scene and HB – Hervey Bay scene (C_m = completeness, C_r = correctness, Q_l = quality, C_{rd} = detection cross-lap rate and C_{rr} = reference cross-lap rate).

Data set	C_m	C_r	Q_l	C_{rd}	C_{rr}
AV1	98.3	99.4	97.7	1.7	27.3
AV2	100	96.2	96.2	0.0	8.0
HB	98.4	98.4	96.9	9.9	7.6
Average	98.9	98	96.9	3.9	14.3

Table 5
Pixel-based evaluation results in percentages for two test data sets: AV1 – first Aitkenvale scene, AV2 – second Aitkenvale scene and HB – Hervey Bay scene (C_{mp} = completeness, C_{rp} = correctness, Q_{lp} = quality, B_f = branching factor and M_f = - miss factor).

Data set	C_{mp}	C_{rp}	Q_{lp}	B_f	M_f
AV1	89.9	94.2	85.2	6.1	11.3
AV2	96.6	96.9	93.7	3.2	3.6
HB	87.6	93.8	82.8	6.6	14.2
Average	91.4	95	87.2	5.3	9.7

were many over-segmented regions. Moreover, the high detection cross-lap rate and miss factor value in the HB data set indicated that there were many under-segmented regions.

4.5. Comparison with other methods

Since different roof extraction methods use different data sets and different evaluation systems and metrics, it is not straightforward to compare the results of the different methods reported in the literature. However, in order to show the current progress of research in the field of 3D building roof extraction, the evaluation results presented by prominent data-driven methods are summarised here. The methods, which employ similar evaluation systems and metrics (completeness, correctness and quality), are then chosen to show the progress made by the proposed method.

While many reported data-driven methods (Vosselman and Dijkman, 2001; Rottensteiner, 2003; Rottensteiner et al., 2004 and Park et al., 2006) lack accompanying objective evaluation results for roof plane reconstruction, others are without results based on the number and area (pixels) of reconstructed planes. For example, Dorninger and Pfeifer (2008) presented results based on the number of buildings whose roof planes were correctly reconstructed, but did not show how many planes per building were correctly reconstructed. Sampath and Shan (2010) showed error statistics based on perpendicular distances from the segmented LIDAR points to the extracted planes. Cheng et al. (2011) and Chen et al. (2006) presented results based on the number of building models, not on the number of planes.

Among the rest of the promising data-driven methods, Khoshelham et al. (2005) evaluated results on 4 simple gable roofs having a total of 10 planes and showed object-based completeness C_m and correctness C_r of 100% and 91%, respectively. The semi-automatic method by Habib et al. (2010) offered C_m = 75% and C_r = 94% in an experiment using 23 buildings comprising 180 planes. Nevertheless, it could not extract planes which are less than 9 m^2 in area. Jochem et al. (2012) evaluated their LIDAR-based roof plane segmentation method on a large data set of 1003 roof planes and achieved C_m = 94.4% and C_r = 88.4%. However, as the authors mentioned, their method had shortcomings. Firstly, it could not extract planes of less than 6 m^2 in area. Secondly, it could not remove very dense vegetation where the slope-adaptive LIDAR echo ratio is high. Moreover, it lost accuracy in the extraction of planes as it used gridded LIDAR data. It is not clear from the reported experimental results whether the methods of Khoshelham et al. (2005) and Habib et al. (2010) function in vegetation areas having trees of similar height to surrounding buildings, or whether the approach of Khoshelham et al. (2005) can extract small roof planes.

While the proposed algorithm is compared with the aforementioned three existing methods, like those of Khoshelham et al. (2005) and Jochem et al. (2012) the proposed method is fully automatic, while the method by Habib et al. (2010) is semi-automatic as it requires a manual monoplotting procedure to delete incorrect

boundaries and add necessary boundary segments. The methods of Khoshelham et al. (2005) and Habib et al. (2010) might offer better planimetric accuracy than the proposed method and that of Jochem et al. (2012), since both use image lines to describe the plane boundaries. However, the proposed method can offer better vertical accuracy than the existing three methods as it uses the raw LIDAR data to describe the planes.

The evaluation results presented in this paper cover 88 buildings consisting of 381 roof planes. The proposed method can extract individual planes as small as 1 m^2, and it applies a new rule-based procedure to remove all kinds of vegetation. Unlike the existing methods, object-based evaluation for the proposed method uses quality, detection and reference cross-lap rates. The latter two metrics indicate the under- and over-segmentation of the input data. In terms of object-based completeness and correctness, the proposed method offered higher performance (C_m = 99% and C_r = 98%) than the three existing methods. Moreover, none of the existing methods showed results using the pixel-based evaluation metrics. In contrast, the proposed method has demonstrated high performance in pixel-based evaluation as well.

5. Conclusion and future work

This paper has presented a new method for automatic 3D roof extraction through an effective integration of LIDAR data and aerial orthoimagery. Like any existing methods, the proposed roof extraction method uses a number of algorithmic parameters, the majority of which are either adopted from the existing literature or directly related to the input data. An empirical study has been conducted in order to examine the sensitivity of the rest of the parameters. It is shown that in terms of object- and pixel-based completeness, correctness and quality, the algorithm performs well when the LIDAR point density decreases, and it successfully removes all vegetation (indicated by similar branching factor in Table 3) even when the LIDAR density is low. However, the over-segmentation (reference cross-lap), and under-segmentation (detection cross-lap) rates increase moderately when the LIDAR density is low.

As compared to three existing methods (Khoshelham et al., 2005; Habib et al., 2010; Jochem et al., 2012), the proposed method can extract planes as small as 1 m^2 and can work in the presence of dense vegetation. The proposed method is fully automatic and experimental results show that it not only offers high reconstruction rates but also can work in the presence of moderate registration error between the LIDAR data and orthoimagery. However, as the registration error grows, so does the likelihood that algorithm will fail to properly extract the roof planes, especially the small planes. The authors plan to test the algorithm on further data sets with large registration errors between the orthoimagery and LIDAR data.

Future work includes rectification of the over- and under-segmentation issue and testing the algorithm on more complex data sets. In order to obtain better planimetric accuracy the research of representing the 3D plane boundaries using the image lines is under investigation. In addition, it will be interesting to test the algorithm on real data with low LIDAR point density.

Acknowledgments

Dr. Awrangjeb is the recipient of the Discovery Early Career Researcher Award by the Australian Research Council (Project Number DE120101778). The authors would like to thank Ergon Energy (www.ergon.com.au) for providing the data sets.

References

Awrangjeb, M., Lu, G., 2008. Robust image corner detection based on the chord-to-point distance accumulation technique. IEEE Transactions on Multimedia 10 (6), 1059–1072.

Awrangjeb, M., Lu, G., Fraser, C.S., Ravanbakhsh, M., 2009. A fast corner detector based on the chord-to-point distance accumulation technique. In: Proc. Digital Image Computing: Techniques and Applications. Melbourne, Australia, pp. 519–525.

Awrangjeb, M., Ravanbakhsh, M., Fraser, C.S., 2010a. Automatic building detection using LIDAR data and multispectral imagery. In: Proc. Digital Image Computing: Techniques and Applications. Sydney, Australia, pp. 45–51.

Awrangjeb, M., Ravanbakhsh, M., Fraser, C.S., 2010b. Automatic detection of residential buildings using LIDAR data and multispectral imagery. ISPRS Journal of Photogrammetry and Remote Sensing 65 (5), 457–467.

Awrangjeb, M., Ravanbakhsh, M., Fraser, C.S., 2010c. Building detection from multispectral imagery and LIDAR data employing a threshold-free evaluation system. International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences 38 (part 3A), 49–55.

Awrangjeb, M., Zhang, C., Fraser, C.S., 2012a. Automatic reconstruction of building roofs through integration of LIDAR and multispectral imagery. In: ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, vol. I-3. Melbourne, Australia, pp. 203–208.

Awrangjeb, M., Zhang, C., Fraser, C.S., 2012b. Building detection in complex scenes thorough effective separation of buildings from trees. Photogrammetric Engineering and Remote Sensing 78 (7), 729–745.

Barista, 2011. The Barista Software. <www.baristasoftware.com.au>.

Chen, L., Teo, T., Hsieh, C., Rau, J., 2006. Reconstruction of building models with curvilinear boundaries from laser scanner and aerial imagery. Lecture Notes in Computer Science 4319, 24–33.

Cheng, L., Gong, J., Li, M., Liu, Y., 2011. 3d building model reconstruction from multi-view aerial imagery and LIDAR data. Photogrammetric Engineering and Remote Sensing 77 (2), 125–139.

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

Foody, G., 2002. Status of land cover classification accuracy assessment. Remote Sensing of Environment 80 (1), 185–201.

Gonzalez, R.C., Woods, R.E., Eddins, S.L., 2003. Digital Image Processing Using MATLAB. Prentice Hall, New Jersey.

Gröger, G., Plümer, L., 2012. Citygml – interoperable semantic 3d city models. ISPRS Journal of Photogrammetry and Remote Sensing 71 (7), 12–33.

Habib, A.F., Zhai, R., Changjae, K., 2010. Generation of complex polyhedral building models by integrating stereo-aerial imagery and LIDAR data. Photogrammetric Engineering and Remote Sensing 76 (5), 609–623.

Jiang, X.Y., Bunke, H., 1994. Fast segmentation of range images into planar regions by scan line grouping. Machine Vision and Applications 7 (2), 115–122.

Jochem, A., Höfle, B., Wichmann, V., Rutzinger, M., Zipf, A., 2012. Area-wide roof plane segmentation in airborne LIDAR point clouds. Computers, Environment and Urban Systems 36 (1), 54–64.

Khoshelham, K., Li, Z., King, B., 2005. A split-and-merge technique for automated reconstruction of roof planes. Photogrammetric Engineering and Remote Sensing 71 (7), 855–862.

Lafarge, F., Descombes, X., Zerubia, J., Pierrot-Deseilligny, M., 2010. Structural approach for building reconstruction from a single DSM. IEEE Transactions on Pattern Analysis and Machine Intelligence 32 (1), 135–147.

Lee, D., Shan, J., Bethel, J., 2003. Class-guided building extraction from IKONOS imagery. Photogrammetric Engineering and Remote Sensing 69 (2), 143–150.

Park, J., Lee, I.Y., Choi, Y., Lee, Y.J., 2006. Automatic extraction of large complex buildings using LIDAR data and digital maps. International Archives of the Photogrammetry and Remote Sensing XXXVI (3/W4), 148–154.

Parmehr, E.G., Zhang, C., Fraser, C.S., 2012. Automatic registration of multi-source data using mutual information. In: ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, vol. I-7. Melbourne, Australia, pp. 301–308.

Rottensteiner, F., 2003. Automatic generation of high-quality building models from LIDAR data. Computer Graphics and Applications 23 (6), 42–50.

Rottensteiner, F., Trinder, J., Clode, S., Kubik, K., 2004. Fusing airborne laser scanner data and aerial imagery for the automatic extraction of buildings in densely built-up areas. In: Proc. ISPRS Twentieth Annual Congress. Istanbul, Turkey, pp. 512–517.

Rottensteiner, F., Trinder, J., Clode, S., Kubik, K., 2005. Using the dempstershafer method for the fusion of LIDAR data and multi-spectral images for building detection. Information Fusion 6 (4), 283–300.

Rutzinger, M., Rottensteiner, F., Pfeifer, N., 2009. A comparison of evaluation techniques for building extraction from airborne laser scanning. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing 2 (1), 11–20.

Sampath, A., Shan, J., 2010. Segmentation and reconstruction of polyhedral building roofs from aerial LIDAR point clouds. IEEE Transactions on Geoscience and Remote Sensing 48 (3), 1554–1567.

Satari, M., Samadzadegan, F., Azizi, A., Maas, H.G., 2012. A multi-resolution hybrid approach for building model reconstruction from LIDAR data. The Photogrammetric Record 27 (139), 330–359.

Song, W., Haithcoat, T., 2005. Development of comprehensive accuracy assessment indexes for building footprint extraction. IEEE Transactions on Geoscience and Remote Sensing 43 (2), 402–404.

Vosselman, G., Dijkman, S., 2001. 3d building model reconstruction from point clouds and ground plans. International Archives of the Photogrammetry and Remote Sensing XXXIV (3/W4), 37–44.

Vosselman, G., Gorte, B.G.H., Sithole, G., Rabbani, T., 2004. Recognising structure in laser scanner point cloud. International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences XXXVI (8/W2), 33–38.