

Automatic Reconstruction of Building Roofs Using LIDAR and Multispectral Imagery

Author

Awrangjeb, M, Zhang, C, Fraser, CS

Published

2011

Conference Title

Proceedings - 2011 International Conference on Digital Image Computing: Techniques and Applications, DICTA 2011

Version

Accepted Manuscript (AM)

DOI

[10.1109/DICTA.2011.69](https://doi.org/10.1109/DICTA.2011.69)

Rights statement

© 2011 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

Downloaded from

<http://hdl.handle.net/10072/99607>

Griffith Research Online

<https://research-repository.griffith.edu.au>

Automatic Reconstruction of Building Roofs Using LIDAR and Multispectral Imagery

Mohammad Awrangjeb, Chunsun Zhang and Clive S. Fraser
Cooperative Research Centre for Spatial Information, University of Melbourne
Level 5, 204 Lygon Street, Carlton Vic 3053, Australia
Phone: +61 3 8344 9182, Fax: +61 3 9654 6515
Email: {mawr, chunsunz, c.fraser}@unimelb.edu.au

Abstract—Automatic 3D reconstruction of building roofs from remotely sensed data is important for many applications including automatic city modeling. This paper proposes a new method for automatic roof reconstruction using LIDAR (LIght Detection And Ranging) data and multispectral imagery. 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 that are used to generate initial roof planes. The structural lines are extracted from the grey-scale version of the orthoimage and they are classified into several classes such as ‘ground’, ‘ground with occasional tree’, ‘tree’, ‘roof edge’ and ‘roof ridge’ using the ground mask, the NDVI image (from the multi-band orthoimage) and the entropy image (from the grey-scale orthoimage). The lines from the latter two classes are primarily used to fit initial planes to the neighbouring LIDAR points. Other image lines within the vicinity of an initial plane are selected to fit the boundary of the plane. Once the proper image lines are selected and others are discarded, the final plane is reconstructed using the selected lines. Experimental results show that the proposed method can handle irregular and large registration errors between the LIDAR data and orthoimagery.

I. INTRODUCTION

Building detection and 3D reconstruction has been an area of active research among the photogrammetric, remote sensing and computer vision communities for at least two decades. While building detection means locating and delineating the building boundaries from the remotely sensed data, such as photogrammetric imagery and height data, building reconstruction means extracting the 3D building information, that includes corners, edges and planes of the building facades, and roofs, and then digitally reconstructing the facades and roofs using the available information. Both building detection and reconstruction have a wide variety of applications including automatic 3D city modeling, disaster management and homeland security [1].

This paper concentrates on the automatic reconstruction of 3D building roofs, for which there are a number of algorithms exist in the literature. These algorithms fall into following three categories [2]. Algorithms in the first category use only the LIDAR (LIght Detection And Ranging) data for 3D reconstruction. However, the success of building reconstruction and the level of detail of the resulting building models using airborne laser scanner data alone depend on the resolution of the LIDAR data, which is most often lower than the

resolution of the accompanying aerial imagery. Moreover, the reconstructed buildings suffer from poor horizontal accuracy. The second category of algorithms use aerial imagery only which is difficult because of problems related to shadows, occlusions and poor contrast. In addition, the extracted information has low vertical accuracy. That is why, algorithms in the third category combine LIDAR data and multispectral aerial imagery in order to overcome the drawbacks of specific sensor types.

In this paper, we propose a new method for automatic reconstruction of building roofs using the raw LIDAR data and multispectral orthoimagery. 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 used to generate initial roof planes. The structural image lines are classified into several classes (‘ground’, ‘ground with occasional tree’, ‘tree’, ‘roof edge’ and ‘roof ridge’) using the ground mask, colour orthoimagery and image texture information. The non-ground LIDAR points near a long roof edge (known as the base line) are used to obtain an initial roof plane. The neighbouring image lines are then used to fit the boundary of the initial roof plane. The image lines that properly fit with the boundary are selected to construct the final roof plane. Once a plane is reconstructed, the neighbouring image lines that are parallel or perpendicular to the base line are used to reconstruct the other planes on the same roof. We provide some experimental results which show that the proposed method can handle irregular and large registration errors between the LIDAR data and orthoimagery.

II. RELATED WORK

Rottensteiner and Briese [3] extract roof planes through a hierarchical segmentation technique based on the statistical behaviour of the surface normal vectors of a DSM (Digital Surface Model) computed from the LIDAR data. Aerial images are integrated to improve the geometric quality of the extracted building models. However, the surface normals in dense data sets are quite noisy [4]. Rottensteiner et al. [2] describe another method comprising three steps. Building segments are first detected from the laser scanning data using the Dempster-Shafer theory for data fusion. In the second step, roof plane detection, the detected building segments are improved using the digital images. Finally, the geometry of the roof plane

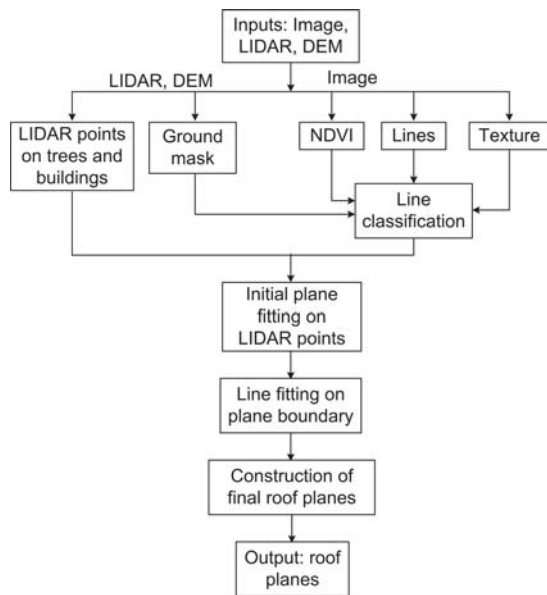


Fig. 1. Proposed building roof reconstruction procedure.

boundaries is improved by matching image edges with the edges of polyhedral building models.

Khoshelham et al. [5] use a split-and-merging procedure based on the height data, with a view to overcoming the problem associated with overgrown and undergrown segmentation of images. In order to avoid losing accuracy due to use of a raster DSM as in [2] and [3], Novacheva [4] works on the raw LIDAR data. The main purpose of integrating a colour image is to refine the building outline. While fusing optical images and DEMs (Digital Elevation Models), Karantzas and Paragios [6] use 3D grammar-based building models to recover the scene's geometry. This model-based approach, though overcomes errors caused by shadows and occlusions, is limited to use for reconstruction of buildings of known shapes only.

III. PROPOSED RECONSTRUCTION PROCEDURE

Fig. 1 shows the overview of the proposed building roof reconstruction procedure. The input data consist of raw LIDAR data, DEM and multispectral orthoimagery. The DEM is only used for an estimation of the ground height while generating the mask from the raw LIDAR data. If it is not available, an approximate DEM can be obtained from the input LIDAR data using the MARS software [7]. The LIDAR points on the buildings and trees are separated as non-ground points. The NDVI (Normalized Difference Vegetation Index) is calculated for each image pixel location using the colour orthoimage. The texture information is estimated at each image pixel location using the grey-scale version of the image. The same grey level image is used to find the image lines that are at least 1m in length. These lines are classified into several classes, namely, 'ground', 'ground with occasional tree', 'tree', 'roof edge' (boundary) and 'roof ridge' (intersection of two roof planes). Lines classified as roof edges and ridges are processed

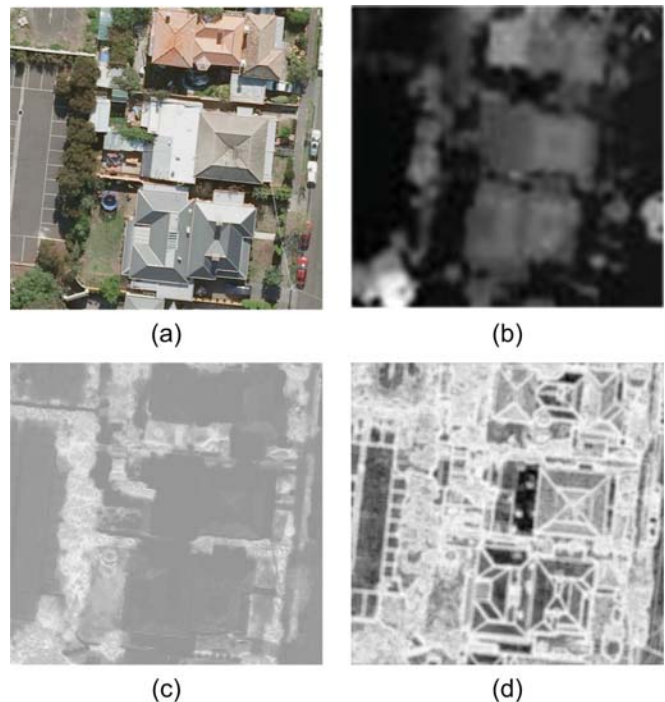


Fig. 2. A test scene: (a) orthoimage, (b) LIDAR data shown as a grey-scale grid, (c) NDVI image from (a), and (d) entropy image from (a).

along with the non-ground LIDAR points while reconstructing a roof plane. An initial plane fit uses the LIDAR points near to a long roof edge, which is considered as the base line for the plane. Local image lines are then fit to the boundary points of the plane. The best fitting lines are finally chosen in order to reconstruct a final roof plane. Other planes on the same building are reconstructed following the same procedure by using the non-ground LIDAR points near to the local image lines, which are either parallel or perpendicular to the base line of an already reconstructed plane.

In the following sections, we first present the data set used for experimentation and then detail the different steps of the proposed roof reconstruction method. The majority of threshold values used in this paper are from the existing literature and the rest will be experimentally validated in a forthcoming paper on the topic.

A. Test Data Set

Fig. 2a presents a test scene which will be used to illustrate the different steps of the proposed reconstruction method. This scene is from Mooney Ponds, Victoria, Australia. Available data comprised of the first-pulse LIDAR data with a point spacing of 1m (Fig. 2b), a DEM (with 1m spacing), and an RGBI colour orthoimage with a resolution of 0.1m.

The orthoimage had been created using a bare-earth DEM, so that the roofs and the tree-tops were displaced with respect to the LIDAR data. Thus, data alignment was not perfect. Apart from this registration problem, there were also problems with shadows in the orthoimage, so the NDVI image, shown

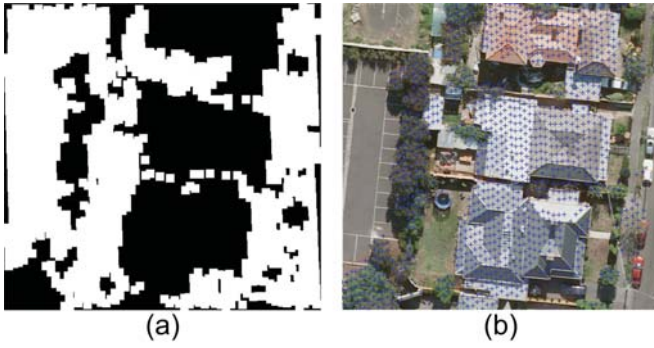


Fig. 3. (a) Ground mask for the scene in Fig 2 and (b) non-ground LIDAR points are overlaid on the orthoimage.

in Fig. 2c, did not provide as much information as expected. Therefore, texture information in the form of entropy [8] (see Fig. 2d) is also employed based on the observation that trees are rich in texture as compared to building roofs. In order to calculate entropy e at a point P within a grey-scale image, a 9×9 subimage I is considered, where P is the centre point. A normalized histogram H for I , involving 256 bins and values between 0 to 1, is formed and entropy is calculated using non-zero frequencies as

$$e = - \sum H_i \log_2 (H_i), \text{ where } 1 \leq i \leq 256 \text{ and } 0 < H_i \leq 1. \quad (1)$$

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 classify roof edges and tree edges while classifying image lines.

B. Mask Generation and Classification of LIDAR points

For each LIDAR point, the corresponding DEM height is used as the ground height H_g . If there is no corresponding DEM height for a given LIDAR point, the average DEM height in the neighbourhood is used as H_g . A height threshold $T_h = H_g + 2.5\text{m}$ [2] is applied to the raw LIDAR height. Consequently, the LIDAR data are divided into two groups: those which reflect from the low height objects such as ground, road furniture, cars and bushes, and those which reflect from the high height objects like buildings and trees.

Each of the LIDAR points in the first set marks white in the ground mask M_g , which is initially a complete black mask. Therefore, as shown in Fig. 3a, M_g indicates the *void areas* where there are no laser returns below T_h , i.e., ground areas covered by buildings and trees. This mask will be used to classify image lines into different classes in Section III-C. We preserve the second set (the non-ground LIDAR points shown in Fig. 3b) to extract the roof plane in Section III-D.

C. Extraction and Classification of Image Lines

In order to extract lines from the grey-scale version of the orthoimage, we first detect edges using the Canny edge detector. We then detect corners on the extracted curves employing a fast corner detector [9]. On each edge, all the

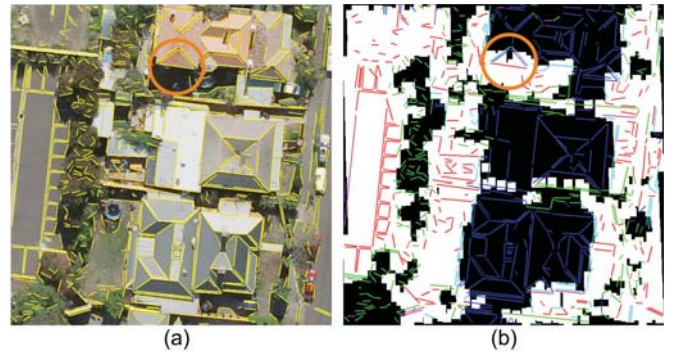


Fig. 4. Image lines: (a) extracted and (b) classified (green: ‘tree’, red: ‘ground’, magenta: ‘ground with occasional tress’, cyan: ‘roof edge’ and blue: ‘roof ridge’).

pixels between two corners or a corner and an endpoint, or two endpoints when enough corners are not available, are considered as a separate line segment. If a line segment is less than 1m in length it is removed. Thus trees having small horizontal areas are removed. Finally, a least-square straight-line fitting technique is applied to properly align each of the remaining line segments. Fig. 4a shows the extracted lines from the test scene.

For classification of the extracted image lines, we consider a rectangular area of 1.5m width on each side of a line. The width of the rectangle is based on the assumption that the minimum building width is 3m. In each rectangle, we estimate the percentage Φ of black pixels from M_g (from Fig. 3a), the average NDVI value Υ (from Fig. 2c) and the percentage Ψ of pixels having high entropy values (from Fig. 2d). We also estimate a binary flag F_b for each rectangle, where $F_b = 1$ indicates that there are continuous black pixels in M_g along the line. Note that in MATLAB [10], a pixel having an entropy value of more than 0.8 is considered as a highly textured pixel because this threshold is roughly the intensity value of pixels along the boundary between the textures.

For a 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 $\Phi \geq 10\%$. If $\Upsilon > 10$ and $\Psi > 30\%$ [11] on either of the sides then the line is classified as ‘tree’. If $\Upsilon \leq 10$, or if $\Upsilon > 10$ but $\Psi \leq 30\%$, then if $F_b = 1$ on both sides then the line is classified as ‘roof ridge’, but if $F_b = 1$ on one side then it is classified as ‘roof edge’. Otherwise ($F_b = 0$ on both sides), the line is classified as ‘ground with occasional trees’, for example, road sides with trees on the nature strip. Fig. 4b shows different classes of the extracted image lines.

Logically, only the lines in classes ‘roof edge’ and ‘roof ridge’ should be considered for reconstruction of roof planes. However, as shown within the orange coloured circle in Fig. 4, due to severe registration error between the LIDAR data and orthoimage, some of the roof edges may be classified as ‘ground’ and some of the roof ridges may be classified as ‘roof edge’. Consequently, though lines classified as ‘roof edge’ or ‘roof ridge’ will be the starting edge while reconstructing a roof plane, lines in other classes may be considered to

complete the plane or to reconstruct a neighbouring plane.

D. Initial Plane Fitting on Non-ground LIDAR Points

We sort the lines in the ‘roof edge’ class in the descending order of their length. Considering the longest roof edge as a base line (see cyan coloured line in Fig. 5a) for the corresponding roof plane, we fit a plane to the non-ground LIDAR points as follows.

$$Ax + By + Cz + D = 0. \quad (2)$$

Let the point spacing between the LIDAR point is denoted as d_f , which is 1m for the test data set. There are 4 unknowns in (2), so we need at least four LIDAR points on the actual plane. In order to find these points we consider a rectangle (width d_f on each side, magenta coloured solid rectangle in Fig. 5a) around the base line. 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 the centre of the building. We do not consider the points which reflect from the wall. The number of such points is small and their height is low while compared to the actual points on the roof. A point within the rectangle is considered to be reflected from the wall if it has high height difference (more than d_f) with majority of the points within the rectangle. Once we have the required number of points on the plane, we construct the initial plane using (2).

In order to find whether the roof plane is flat or has a upward or downward slope with respect to the base line, we find the nearest and the farthest points P_n and P_f from the base line. If P_n and P_f have similar height the plane is flat. If the height at P_n is smaller than that at P_f then the plane has a upward slope. Otherwise, it has a downward slope with respect to the base line.

The fitted plane is then iteratively extended towards the building centre by considering a rectangle of width d_f outside the previous rectangle (see magenta coloured dotted rectangle in Fig. 5a). As long as, at least 1 point within the new rectangle is compatible (slope and residue of the plane do not change much) the plane is extended. At each iteration we update P_n and P_f in order to avoid extending the plane on a neighbouring plane. We also exclude the points reflected from the wall. Similarly, we extend the plane towards the left and right of the base line as long as the residue of the plane does not increase. Fig. 5b shows the fitted plane as a solid magenta coloured rectangle. The points reflected from the wall are shown in red circles and those reflected from the roof plane are shown in either blue coloured squares (boundary of the initial roof plane) or green coloured circles (inside the initial plane boundary).

To obtain the boundary points of the plane we construct the Delaunay triangulation among the fitted points (see cyan coloured triangles in Fig. 5b). A triangle side that connects the two neighbouring points and constitutes only one triangle should be on the boundary of the plane. However, a triangle side that connects two non-neighbouring points but constitutes only one triangle is not on the boundary of the plane. To

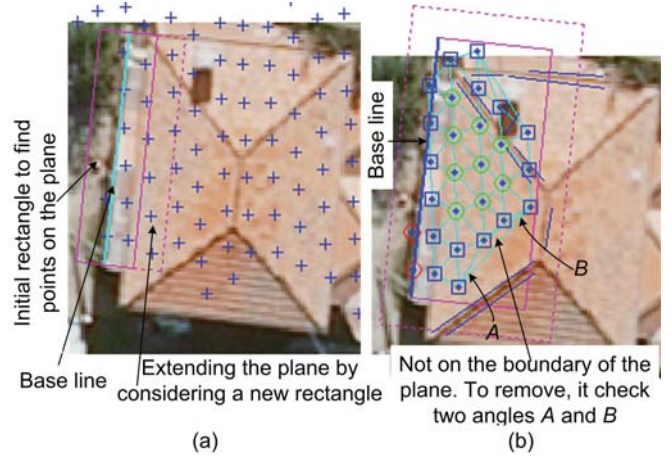


Fig. 5. (a) Fitting a plane to the LIDAR points and (b) finding the boundary points of the plane.

remove such a triangle side we check the two inside angles A and B , as shown in Fig. 5b. If any of the angles is less than 40° (ideally, 45°), then the side of the triangle is removed. As a result, two other sides of that triangle become the candidates to be on the boundary of the plane. The corresponding two triangles are then checked to verify their candidacy. This procedure continues until all the outside triangles in the triangulation have angles near to 45° each. This constraint ensures that the outside triangles connect the neighbouring points only and their sides, each of which constitutes only one triangle, provides us the boundary points of the plane.

E. Image Line Fitting on the Plane Boundary

Since there is registration error between the LIDAR points and orthoimage, the initial plane fitted with the LIDAR points in the previous section may not directly correspond to the image lines on the actual roof plane. Therefore, we extend the solid magenta coloured rectangle in Fig. 5b by d_f on each side and find all the image lines within this extended rectangle. The base line is also updated with the longest line which is parallel to and within a neighbourhood (initial rectangle shown in Fig. 5a) of the previous base line. In Fig. 5b, the extended rectangle is shown as a dotted magenta coloured rectangle, the base line is shown as a thick blue coloured line and the other image lines are shown as thin blue coloured lines.

Due to quantization errors, the extracted image lines may not be aligned properly with respect to the base line. Furthermore, we may extract two lines for the same roof ridge as shown in a dotted ellipse in Fig. 6a. We find the lines which are parallel (within $\pm\frac{\pi}{8}$) or perpendicular (within $\pm\frac{\pi}{2} \pm \frac{\pi}{8}$) to the base line and adjust them [1]. In addition, if there are two parallel lines found for the same image line (within a neighbourhood of 5 image pixels) then we obtain an average line for these two. Fig. 6a shows the adjusted and average lines in red colour.

Because of irregular registration error, each of the image lines obtained as above may be inside, outside or intersected with the boundary points (see Fig. 6a). We set the following

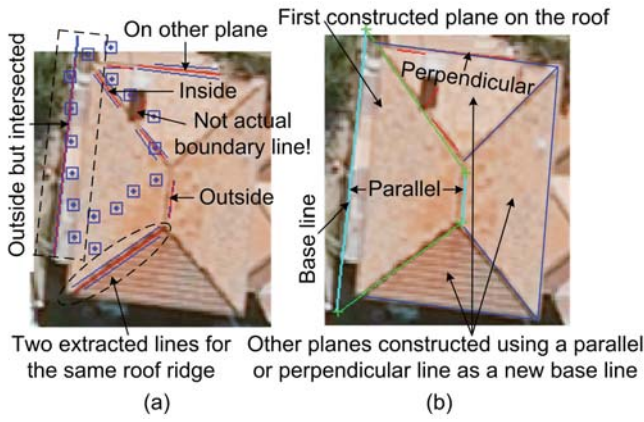


Fig. 6. (a) Fitting image lines to an initial plane and (b) constructed final planes.

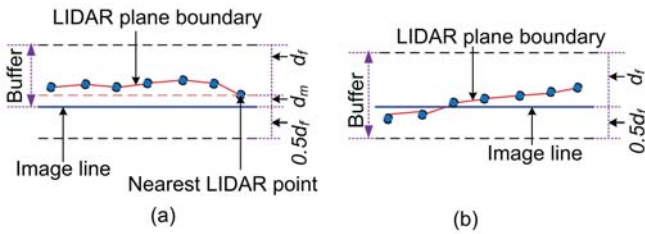


Fig. 7. Fitting LIDAR plane boundary with an image. The line is: (a) completely inside or outside of the boundary and (b) intersected with the boundary.

rules to categorize these image lines. For an image line, we divide the boundary points into two groups: *inside* and *outside*. Now with respect to plane boundary the line is:

- *completely outside*: All the points will be in the *inside* group, but no points in the *outside* group.
- *completely inside*: There will be no points in the *inside* group within $\frac{d_f}{2}$ distance of the line, but in the *outside* group there will be at least two points within d_f distance of the line.
- *outside but intersected*: There will be only a few points in the *outside* group and they mainly reside within $\frac{d_f}{2}$ distance of the line. However, the majority of the boundary points fall into the *inside* group and many of them stay within d_f distance of the line.
- *inside but intersected*: In the *outside* group there will be only a few points within $\frac{d_f}{2}$ distance of the line. But in the *inside* group there are many points within d_f distance of the line.

The first two cases are illustrated by Fig. 7a and the other two by Fig. 7b. In fact, some of the lines may not be on the actual plane boundary or may be on the neighbouring planes, as shown in Fig. 6a. Such lines can be avoided when we count the number of actual fitted boundary points and select only those lines which have high number of fitted points. The fitted points can be easily obtained within the buffer, as shown in Fig. 7. For a line which is completely outside or inside of the initial plane boundary, the buffer is situated on the side

of the line where the corresponding boundary segment is and its width is $d_m + d_f$, where d_m is the perpendicular distance from the nearest LIDAR boundary point to the base line (see Fig. 7a). For an intersected line, the buffer is situated on both sides of the line and its width is $1.5d_f$. As shown in Fig. 7b, the two-third of the buffer is situated on the side of the line where the majority of the intersected boundary segment is.

If there are no or only one fitted point within the defined buffer of a line, then the line is removed. The line which is shown on a neighbouring plane in Fig. 6a is thus removed. For each fitted line, we find the two end LIDAR points from the fitted boundary segment.

F. Construction of the Final Roof Plane

Now we have the image lines with the fitted boundary points. The lines which are parallel to the base line and have at least two fitted points are the first selected lines (shown in cyan colour in Fig. 6b) on the final plane boundary.

Then we sort the remaining lines (shown in green and red colours in Fig. 6b) in descending order of the number of fitted points. We select the line which has the maximum number of fitted points and find the intersections (shown in green cross signs in Fig. 6b) with the already selected lines. We record the ends of the selected lines which have got the new end (intersected) points. The procedure continues for the rest of the sorted lines. If an end gets two intersection points we keep the earliest one and discard the new one. If both of the intersected points of a newly selected line are discarded this line is also discarded. In Fig. 6b, the selected lines are shown in green and discarded lines in red.

Due to missing of an image line, there may be ends which do not get any intersected points. In such cases, we fit a straight line (using the least square technique) to the LIDAR boundary points that reside between the two ends. This straight line is added as a selected line and the intersection points are assigned to the previously unassigned ends. Latter when the neighbouring plane is constructed, this line is replaced with the intersected line of the two planes.

In order to assign height to an intersection point, we check the two corresponding LIDAR end points (of the intersecting lines). If they are the same point, then its height is assigned to the intersection point. If they are different then the two corresponding lines are checked. If both of them were considered as outside (*completely outside* or *outside but intersected*) lines in Section III-E, the height of the nearest LIDAR end point is assigned. Otherwise, the height of the farthest LIDAR end point is assigned.

Once a plane has been reconstructed, we find all the local image lines which are parallel or perpendicular to the base line of the constructed plane and they get priority than the other long lines to reconstruct the neighbouring planes in the next iteration started in Section III-D. As shown in Fig. 6b, these lines are used as the new base lines while reconstructing the neighbouring planes. Fig. 6b shows the other reconstructed planes in blue coloured lines. This way all the planes of a

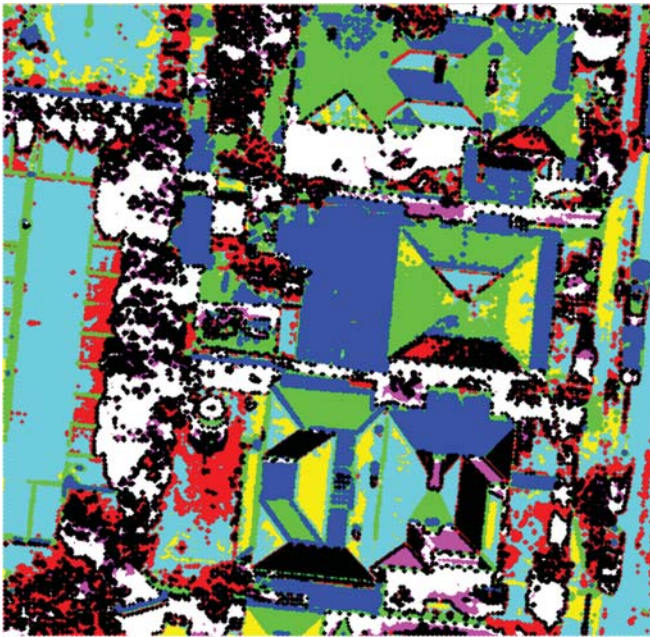


Fig. 8. Colour segmentation of the image in Fig. 2a.

building are reconstructed first before the reconstruction of planes of another building.

IV. CONCLUSION AND FUTURE WORK

Automatic reconstruction of building roof is important for many applications including automatic city modeling. This paper has proposed a new method for automatic roof reconstruction by integrating the LIDAR data and photogrammetric imagery. The LIDAR data is divided into two groups: ground and non-ground points. The non-ground points are used to generate the initial roof planes. The structural lines from the orthoimage are classified into several classes. The lines in the ‘roof edge’ and ‘roof ridge’ classes are primarily used to fit the initial planes and thereby for constructing the final roof planes. The experimental results show that the proposed method is capable of handling large and irregular registration error between the two data sources.

However, this work is still under investigation and a comprehensive experimental validation as well as the objective evaluation on different data sets are essential to prove the complete effectiveness of the proposed method. In addition, when the density of LIDAR points is low, there may not be sufficient number of LIDAR points on a small roof plane to construct an initial plane. In such a case, colour segmentation of the image can be used to reconstruct the missing planes. Fig. 8 shows colour segmentation results using K-means clustering algorithm. We see that the segmentation results is quite good and small planes can be distinguished from the neighbouring planes. The future work include all the above issues including further improvement of the proposed reconstruction method.

ACKNOWLEDGMENT

The authors would like to thank the Department of Sustainability and Environment (www.dse.vic.gov.au) for providing the LIDAR data and orthoimagery for the Mooney Ponds test site.

REFERENCES

- [1] M. Awrangjeb, M. Ravanbakhsh, and C. S. Fraser, “Automatic detection of residential buildings using lidar data and multispectral imagery,” *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 65, no. 5, pp. 457–467, 2010.
- [2] F. Rottensteiner, J. Trinder, S. Clode, and K. Kubik, “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, 2004, pp. 512–517.
- [3] F. Rottensteiner and C. Briese, “Automatic generation of building models from lidar data and the integration of aerial images,” in *Proc. ISPRS working group III/3 workshop on 3-D reconstruction from airborne laserscanner and InSAR data*, vol. IAPRS XXXIV, Part 3/W13, Dresden, Germany, 2003, pp. 512–517.
- [4] A. Novacheva, “Building roof reconstruction from lidar data and aerial images through plane extraction and colour edge detection,” in *Proc. ISPRS Twenty First Annual Congress*, vol. IAPRS XXXVII, Part B6b, Beijing, China, 2008, pp. 53–58.
- [5] K. Khoshelham, Z. Li, and B. King, “A split-and-merge technique for automated reconstruction of roof planes,” *Photogrammetric Engineering and Remote Sensing*, vol. 71, no. 7, pp. 855–862, 2005.
- [6] K. Karantzas and N. Paragios, “Large-scale building reconstruction through information fusion and 3-d priors,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 48, no. 5, pp. 2283–2296, 2010.
- [7] *MARS Explorer*, Merrick & Company, Aurora, CO, USA, 2011, version 7.0.
- [8] R. C. Gonzalez, R. E. Woods, and S. L. Eddins, *Digital Image Processing Using MATLAB*. New Jersey: Prentice Hall, 2003.
- [9] M. Awrangjeb, G. Lu, C. S. Fraser, and M. Ravanbakhsh, “A fast corner detector based on the chord-to-point distance accumulation technique,” in *Proc. Digital Image Computing: Techniques and Applications*, Melbourne, Australia, 2009, pp. 519–525.
- [10] *MATLAB – The Language of Technical Computing*, The Mathworks, Natick, MA, USA, 2010, release R2010a.
- [11] M. Awrangjeb, M. Ravanbakhsh, and C. S. Fraser, “Automatic building detection using lidar data and multispectral imagery,” in *Proc. Digital Image Computing: Techniques and Applications*, Sydney, Australia, 2010, pp. 45–51.