

A Triangulation-based Technique for Building Boundary Identification from Point Cloud Data

Mohammad Awrangjeb and Guojun Lu
School of Engineering and Information Technology
Federation University Australia
Gippsland Campus, Churchill Vic 3842

Email: {Mohammad.Awrangjeb,Guojun.Lu}@federation.edu.au

Abstract—Building boundary identification is an essential prerequisite in building outline generation from point cloud data. In this problem, boundary edges that constitute the building boundary are identified. The existing solutions to the identification of boundary edges from the input point set have one or more of the following problems: ineffective in finding appropriate edges in a concave shape, incapable of determining a ‘hole’ or ‘concavity’ inside the shape separately, dependant on additional information such as the scan direction that may be unavailable, and incompetent in determining the boundary of a point set from the boundaries of two or more subsets of the point set. This paper proposes a new solution to the identification of building boundary by using the maximum point-to-point distance in the input data. It properly detects the boundary edges for any type of shape and separately recognises holes, if any, inside the shape. The unique feature of the proposed solution is that it can identify the boundary of a point set from the boundaries of two or more subsets of the point set. It does not require any additional information other than the input point set. Experimental results show that the proposed solution can preserve details along the building boundary and offer high area-based completeness and quality, even in low density input data.

I. INTRODUCTION

A point cloud is a set of data points in a given coordinate system. It is usually defined in 3D by x , y and z coordinates and intended to represent the external surface of an object. LIDAR (LIght Detection And Ranging) is an optical remote-sensing technique that uses laser light to densely sample the surface of the earth, producing highly accurate x , y and z measurements. LIDAR, for example, airborne laser scanning technology, is an emerging cost-effective alternative to traditional surveying techniques such as photogrammetry. It produces mass point cloud data sets that can be managed, visualized and analyzed using computers. Fig. 1 shows a sample of LIDAR point cloud data superimposed on the aerial image.

LIDAR technology has several advantages while compared to its photogrammetric counterpart. Firstly, LIDAR system provides highly accurate 3D measurements. Secondly, high density cost effective point cloud data are now available. High density data not only offer high accuracy in 3D measurements, but also assist in extraction of detail on the building roof, e.g.,



Fig. 1. A sample of LIDAR point cloud data superimposed on the aerial image. Points are shown in colours based on high to low height values: red (high), orange, yellow (medium), green and light blue (low).

chimneys. Finally, LIDAR point cloud data has accelerated the automation of object extraction methods.

Automatic building extraction from LIDAR point cloud data has got a significant attention in the last decade. It is important for various real world applications such as urban modelling, disaster management and homeland security. Building extraction using LIDAR point cloud data essentially involves a segmentation technique where each building is identified as a subset of the input point set. The individual building boundary must then be identified in order to latter represent the building as a regular polygon known as the building footprint. The investigation reported in this paper concentrates on the boundary identification process, where boundary edges that form the boundary are identified. Each edge is a line that connects two consecutive boundary points.

A Delaunay triangulation is first formed for the input point cloud and an initial building boundary (in terms of edges) is formed using the outside edges of the triangulation. The initial boundary is an over estimation of the actual boundary and does not show concavities. Then, the edges in the initial boundary are gradually replaced with the inside edges, which reside close to the current boundary, based on the maximum point-to-point distance in the input data. The refined boundary (still in terms of edges) is then traced to generate a series of points that define the building boundary.

Once the outside boundary of an input shape is obtained, the boundary of any inside ‘holes’ or ‘cavities’, usually seen if present as large gaps within the input data, can also be obtained. Long edges inside the triangulation are removed

to locate such holes. Then the boundary of each hole is determined by using the same procedure applied to obtain the outside boundary.

An important and unique property of the proposed boundary identification algorithm is that it can extract the boundary of a large point set from the boundaries of two or more of its point subsets. Any inside hole can also be determined from the boundaries of subsets. This is particularly useful for obtaining a building boundary from the boundaries of its planes.

During evaluation, in order to avoid any bias that may have been introduced in the point cloud segmentation, the point set inside each reference building is directly provided as input to the proposed building outline generation. The accuracy of the extracted boundary with respect to the reference building is estimated using area-based evaluation metrics.

II. RELATED WORK

Building boundary extraction methods from LIDAR data can be categorised into two groups – direct and indirect extraction. In the direct group, the boundary is directly extracted from the input point set, as in the convex hull algorithm. In the indirect group, the input point set of a building is first used to generate a binary image (represented by a grid of cells), where the background represents the ground and the foreground represents the building. Then the boundary is extracted using an edge detector around the foreground, or the boundary of the foreground cells is directly accepted as the building footprint. Sometimes, the input points close to the extracted boundary replace the edge points.

Among the indirect methods, Awrangjeb and Fraser [1] converted the point cloud into a binary mask (0.25 m resolution) and used the Canny edge around the black shape in the mask as the building boundary. In a second method, they used a mask resolution of 1 m and replaced the Canny edge points by the nearest LIDAR points. Haithcoat et al. [2] used a Gaussian kernel to extract building boundaries from a normalised-DSM (Digital Surface Model) of 1.3 m resolution.

In the direct method, for a given point set P in the Euclidean space, the determination of an optimal boundary that surrounds all points in P is designated as a non-trivial problem because of the possible presence of concavities in the object, e.g., an L-shape building. The determination of the convex hull or convex envelope, which is the smallest convex region enclosing all the points in P , via Computational Geometry does not help much. Thus, Jarvis [3] modified the convex hull formation algorithm by limiting the search space for the next boundary point within a circular neighbourhood of the current boundary point. Consequently, the modified algorithm allows some concavities in the extracted shape. However, this algorithm was not found to be particularly useful due to an uneven point distribution in the point cloud. In turn, the circular neighbourhood was replaced with a rectangular neighbourhood, whose dimensions were set at least twice the maximum point spacing in and across the scan direction [4]. However, this algorithm still does not work when the input data is noisy or has large gaps along the periphery. It also cannot identify separate boundaries for

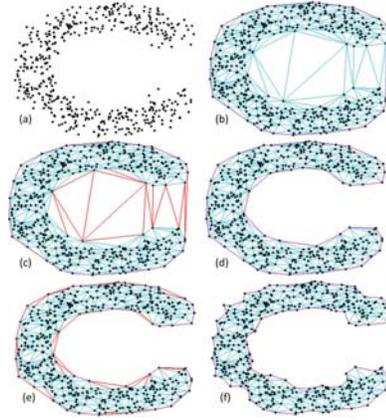


Fig. 2. Boundary extraction: (a) input point cloud P , (b) Delaunay triangulation and the initial boundary (magenta colour), (c) removal of long boundary edges that are shown in red colour, (d) boundary after removal of long edges, (e) removal of short boundary edges that are shown in red colour and (f) final boundary after removal of long and short edges.

multiple objects present in the data either. Moreover, it requires the scan direction information in order to set the orientation of the rectangular neighbourhood window.

The main problem with the above algorithms, including the convex-hull algorithm, is that an inner boundary (hole inside a shape) of a given point set cannot be extracted. Edelsbrunner et al. [5] proposed the so-called α -shape determination algorithm that is able to extract both the inner and outer boundary of a shape. This algorithm has been extensively applied in various applications for the last three decades. However, it is incompetent in determining the boundary of a point set from the boundaries of two or more subsets of the point set. The determination of a boundary from two or more subset boundaries is important, since the modern LIDAR systems are capable of capturing very high density point cloud data. Consequently, there may be millions of points in a small geographic area, for example, on a block of residential buildings. Thus, this will be cost effective if the boundary of a building boundary can be obtained from the boundaries of its planes, since the points within a plane boundary are only a small fraction of the original input point set.

III. PROPOSED BOUNDARY IDENTIFICATION METHOD

For a given point cloud P , the proposed boundary identification technique first obtains the outside boundary. If there are two or more disconnected objects in P , the algorithm can find the boundary for each individual object. It can then extract the boundary of a ‘hole’ or ‘concavity’, if any, inside the point set. An important and unique property of the proposed technique is that it can extract the boundary of a large point set from the boundaries of two or more of its point subsets. Any inside hole can also be determined from the boundaries of subsets. This is particularly useful for obtaining a building boundary from the boundaries of its planes. The proposed boundary identification algorithm is described below using a number of shapes including the C-shape shown in Fig. 2(a).

A. Outside Boundary Identification

A Delaunay triangulation is formed for P . As shown in Fig. 2(b), every side of a triangle inside the triangulation is associated with exactly two neighbouring triangles. However, for a triangle along the periphery of P , one of the sides is associated with only one triangle. Fig. 3a shows such a triangle ABC , where BC is not associated with any other triangle except $\triangle ABC$. Thus, BC is an edge along the boundary. All the edges such as BC form the initial boundary. The initial boundary of P is shown in magenta in Fig. 2b.

Some of the edges in the initial boundary connect the boundary points that are far away from each other, i.e., they connect non-neighbouring points. Such edges should be removed in order to reveal any concavities present in the shape or if two or more objects in P are to be disconnected in order to obtain individual boundaries. Let d_{max} indicate the maximum point-to-point distance in P , i.e., there is at least one neighbouring point $q \in P$ within d_{max} of a point $p \in P$. If a boundary edge BC is longer than $2d_{max}$ (threshold $T_d = 2d_{max}$), then BC is removed, which means $\triangle ABC$ no longer exists. The value of d_{max} is 0.5 m for the shape in Fig. 2. The removal of BC thus leaves each of AB and AC to be now associated with only one triangle. Consequently, AB and AC become two new boundary edges replacing BC in the initial boundary and are separately tested if each of them connects two neighbouring points. The removal of long boundary edges iteratively continues until every edge along the boundary is at most T_d in length. Fig. 2c shows the edges to be removed in red.

The boundary after removal of long edges is shown in Fig. 2d. As can be seen, this boundary can be used to represent an approximate boundary of the input point cloud. However, if further detail (i.e., small concavities) along the boundary is required, a refinement of the current boundary is possible. The proposed refinement below can also reveal a narrow path from outside to a large concavity inside the current boundary.

Let $\triangle ABC$ be along the current boundary and BC be a boundary edge. Let θ be the opposite angle of BC , as shown in Fig. 2a. If θ is close to a straight angle (within $\frac{\pi}{16}$), or if the length of BC is more than d_{max} and θ is an obtuse angle, then replace BC with AB and AC in the boundary. In turn, $\triangle ABC$ no longer exists and each of AB and AC is now associated with only one triangle. Consequently, AB and AC along with their opposite angles in the corresponding triangles are separately checked to ascertain if any can be removed by using the same test. The removal of boundary edge iteratively continues until none of the edges can be removed using the above test. Fig. 2e shows the newly removed edges in red and Fig. 2f shows the refined boundary in magenta.

Fig. 4a shows a point cloud for a composite MA-shape ($d_{max} = 1.3$ m). Fig. 4b shows the initial boundary formed from the Delaunay triangulation. Figs. 4c-d show the extracted boundaries after removal of long and both long and short edges, respectively.

Fig. 5 shows the same for a rectangular shape with a

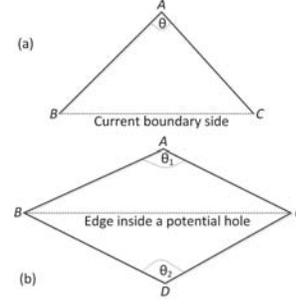


Fig. 3. Delaunay triangles: (a) a triangle along the current boundary and (b) two triangles forming a potential inside ‘hole’ or ‘concavity’.



Fig. 4. Boundary extraction for a composite MA-shape: (a) input point cloud P , (b) Delaunay triangulation and the initial boundary (magenta colour), (c) boundary after removal of long edges and (d) boundary after removal of long and short edges.

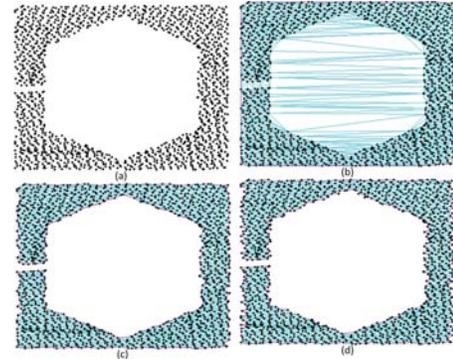


Fig. 5. Boundary extraction for a rectangular shape with an inside hole and tunnel: (a) input point cloud P , (b) Delaunay triangulation and the initial boundary (magenta colour), (c) boundary after removal of long edges and (d) boundary after removal of long and short edges.

hexagonal concavity inside it ($d_{max} = 1.3$ m). The concavity can be reached from outside by a narrow tunnel on the left side of the shape. The width of the tunnel is 2.5 m.

The detail after removal of short edges along the boundary is more noticeable in a curved shape (see example in Figs. 2d-f) than in a straight-edged shape (see example in Figs. 4c-d and 5c-d). This is because the consecutive boundary points along a curved side may reside away from each other and, thus, the angle θ may be large.

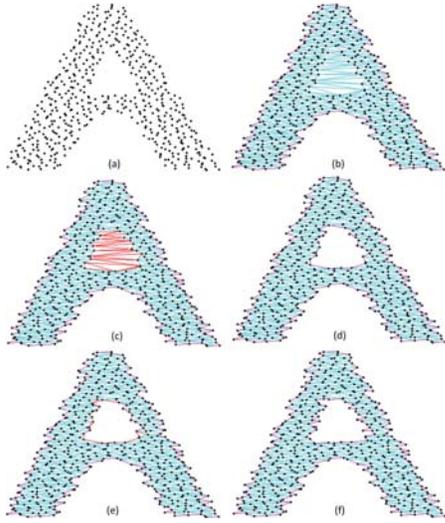


Fig. 6. Inside hole extraction of A-shape: (a) input point cloud P , (b) outside boundary, (c) inside long edges shown in red (d) hole after removal of long edges, (e) inside short edges near the hole are shown in red and (f) hole after removal of long and short edges.

B. Inside Boundary (Hole) Identification

In order to find an inner cavity within a shape, for example, the hole of A-shape in Fig. 6a, the same procedure can be applied. However, in this case the edges inside the boundary are tested instead of those along the boundary.

After obtaining the boundary edges (see Fig. 6b) using the procedure discussed above, the boundary edges are not considered anymore, but all the edges inside the boundary are tested if they are within one or more potential holes. If an inside edge longer than T_d is found, then it is removed and the remaining four sides of the two triangles on both the sides are further tested. Fig. 3b shows an inside edge BC between $\triangle ABC$ and $\triangle BDC$. If $|BC| > T_d$ then it is considered to be inside a hole and, thus, is removed. A hole inside the shape is now found. The four other edges of $\triangle ABC$ and $\triangle BDC$ are also tested separately to ascertain if any of them is longer than T_d and can be removed. When a long edge is removed the inner cavity becomes larger. Consequently, the iterative procedure removes all the long edges as shown by red lines in Fig. 6c. Fig. 6d shows the boundary edges of the hole after removal of long edges.

Any detail along the boundary of the hole can now be obtained by applying the same angle constraint discussed earlier. Fig. 6e shows the removed edges in red and Fig. 6f shows the final hole boundary.

C. Boundary from Point Subsets

An important property of the proposed boundary extraction algorithm is that it can extract the boundary of a large point set P from the boundaries of two or more subsets of P . Any hole inside P can also be obtained separately from the boundaries of subsets. This is particularly useful for obtaining a building boundary from the boundaries of its planes. The computational

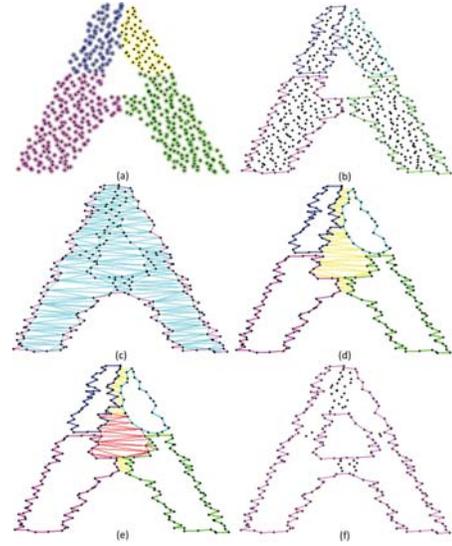


Fig. 7. Boundary from point subsets: (a) input point cloud P decomposed into 4 subsets, (b) four boundaries of point subsets, (c) outside boundary of P from its 4 subset boundaries, (d) inside edges within the gap of subset boundaries in yellow colour, (e) removed inside edges in red colour and (f) final hole and outside boundaries.

cost is also reduced since now the Delaunay triangulation is formed on small subsets of P .

As shown in Fig. 7a, the A-shape has been decomposed into 4 small shapes. The four boundaries of the subsets are shown in Fig. 7b. In order to find the outside boundary of the A-shape, only the points on these four boundaries are now given as the input to the boundary extraction algorithm. Let the input point set be $P_s \subset P$. There is no change required in the procedure discussed in Section III-A, since the algorithm for extraction of the outside boundary works only on the outside edges. Fig. 7c shows the resulting outside boundary in pink colour and the inside edges in cyan colour.

Let the set of inside edges in Fig. 7c be E_i . If there is a hole inside the point set of a subset, that hole can be extracted using the algorithm in Section III-B. Only the edges inside the subset boundary are considered. However, if there is a hole among the subset boundaries, as in Fig. 7, the algorithm for obtaining the inside hole from the original point set can be applied with the following modification, which is necessary in order to keep the subset boundaries unchanged as well as to effectively extract the hole among the subset boundaries.

From E_i , those edges that reside inside any of the subset boundaries, are removed. Fig. 7d shows the remaining inside edges in yellow colour and the subset boundaries in 4 different colours. The hole extraction algorithm in Section III-B is now applied to the remaining edges. Fig. 7e shows the edges in red colour that have been removed. As with the original algorithm the modified hole extraction algorithm does not remove the edges that are already on any of the subset boundaries or on the extracted outside boundary. Fig. 7f shows the extracted inside as well as the outside boundaries.

IV. PERFORMANCE STUDY

In the performance study conducted to assess the proposed approach, six scenes from four data sets from four locations were employed. The detail of the evaluation results has been discussed and analysed.

TABLE I

SCENES IN DATA SETS: AV = AITKENVALE, HB = HERVEY BAY, VH = VAIHINGEN, KN = KNOX. PD = POINT DENSITY IN POINTS/M², d_{max} = MAXIMUM POINT-TO-POINT DISTANCE IN METRE, N_B = NUMBER OF BUILDINGS, N_C = NUMBER OF CORNERS.

Scenes	PD	d_{max}	N_B	N_C
AV	26.1	0.20	5	54
HB	12.3	0.50	22	161
VH 1	3.8	1.00	22	353
VH 2	4.2	1.00	11	216
VH 3	3.7	1.00	44	444
KN	1.1	1.20	50	430

A. Data Sets

Table I shows the characteristics of six test scenes from four data sets (more information is in [6]). As can be seen, the data sets have been chosen with varying point density from high to low. While a high density point cloud provides the opportunity to extract small detail along the building boundary, a low density point cloud shows difficulty even in identification of proper building boundary for a medium size building.

For all data sets, 2D reference data were created by monoscopic image measurement using the Barista software [7]. All visible buildings were digitized as polygons irrespective of their size. The reference data included garden sheds, garages, etc. These were sometimes as small as 3 m² in area.

The Aitkenvale (AV) has a scene of five large buildings. The Hervey Bay (HB) data set also has one scene but it contains many small buildings, e.g., 6 were between 8 to 11 m² in area. The Vaihingen (VH) data set has three test scenes with slightly varying point density. There were also some small reference buildings in these scenes (2 buildings were between 3 to 10 m² in VH 1; 2 between 4 to 15 m² in VH 2; and 5 between 8 to 15 m² in VH 3). In the KN data set there were the most number of small buildings among all test data sets: 5 buildings were between 3 to 10 m² and 8 buildings were between 10 to 20 m² in area. Some buildings were not included in the reference set either due to absence of enough LIDAR points within the small building area or missing of the point cloud data.

In order to find the input point data P for a reference building, all the LIDAR points within the reference building were included in P . However, due to occasional missing of points in LIDAR data or because of misalignment between the orthoimage and LIDAR data, there was some gaps in between each pair of reference and extracted boundary.

B. Evaluation System and Indices

Each extracted building boundary is evaluated with respect to its reference boundary using an area-based (also known



Fig. 8. Extracted boundaries (in yellow) from LIDAR point cloud are shown on the aerial image for the Aitkenvale data set.

as pixel-based) evaluation. A grid G of pixels at 0.5 m resolution is generated. The numbers of true positive pixels TP_p , false positive pixels FP_p and false negative pixels FN_p are counted and completeness C_m , correctness C_r , quality Q_l , area omission error A_{oe} and area commission error A_{ce} are estimated according to [1].

The results presented below are averaged over all buildings for each data set separately. The higher values for C_m , C_r , Q_l show the better performance. In contrast, the lower values for A_{oe} and A_{ce} indicate the better performance.

TABLE II

AREA-BASED EVALUATION RESULTS OF THE EXTRACTED BOUNDARY BY THE PROPOSED ALGORITHM: AV = AITKENVALE, HB = HERVEY BAY, VH = VAIHINGEN, KN = KNOX. C_m = COMPLETENESS, C_r = CORRECTNESS, Q_l = QUALITY, A_{oe} = AREA OMISSION ERROR AND A_{ce} = AREA COMMISSION ERROR IN PERCENTAGE.

Scenes	C_m	C_r	Q_l	A_{oe}	A_{ce}
AV	96.6	100	96.2	3.4	0
HB	94.1	100	93.7	5.9	0
VH 1	92.4	99.5	92.0	7.6	0.6
VH 2	92.6	99.6	92.3	7.4	0.4
VH 3	89.2	99.7	88.6	10.9	0.3
KN	75.5	99.6	74.7	24.5	0.4
Average	90.1	99.7	89.6	10.0	0.3

C. Evaluation Results and Discussions

Fig. 8 shows the boundaries identified from LIDAR point cloud data of the AV data set.

Table II shows the evaluation of the extracted building boundaries for the test scenes. For the extracted boundaries, the correctness values were almost 100% because the input point sets were exactly within the reference building boundaries. The observation is also supported by close to zero area commission errors for all test data sets. However, the completeness values were smaller than the correctness values because pixels along the reference building footprints were not in the extracted building boundaries due to low point point density compared to the reference pixel resolution (0.5 m). They decreased with the decrease of input point density. Although the best completeness was found for the AV data

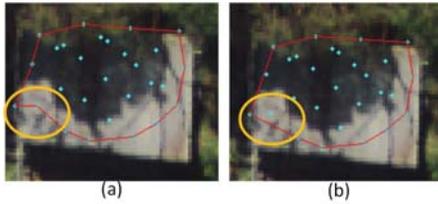


Fig. 9. Example of boundary extraction from a building in the KN scene by: (a) α -shape and (b) proposed algorithm.

set due to its high point density, the worst performance was observed in the KN data set since the extracted boundaries were determined much smaller than the original size due to its low point density. This fact is also evident from the lowest and highest area omission errors for the AV and KN data sets, respectively.

TABLE III

AREA-BASED EVALUATION RESULTS OF THE EXTRACTED BOUNDARY BY THE α -SHAPE ALGORITHM [5]: AV = AITKENVALE, HB = HERVEY BAY, VH = VAIHINGEN, KN = KNOX. C_m = COMPLETENESS, C_r = CORRECTNESS, Q_l = QUALITY, A_{oe} = AREA OMISSION ERROR AND A_{ce} = AREA COMMISSION ERROR IN PERCENTAGE.

Scenes	C_m	C_r	Q_l	A_{oe}	A_{ce}
AV	96.4	100	96.1	3.6	0
HB	92.6	100	92.3	7.4	0
VH 1	90.9	99.6	90.4	9.1	0.4
VH 2	91.7	99.7	90.8	8.3	0.3
VH 3	87.4	99.8	86.6	12.6	0.2
KN	72.1	99.8	71.0	27.9	0.2
Average	88.5	99.8	87.9	11.5	0.2

D. Comparative Results

The boundary extraction technique in [4] fits a rectangular neighbourhood with respect to the scan direction of the point cloud. This means for a given point cloud when the information related to the scan direction is not available, the rectangular neighbourhood becomes a circular neighbourhood used in its predecessor [3]. Moreover, similar to other convex hull algorithms, this technique fails to extract any holes that may present inside a shape. Since, the input data in the experiment did not come with the information relevant to the scan direction, in order to compare the proposed boundary identification procedure, only one existing technique, the well known and widely applied α -shape algorithm, was implemented [5]. The value for α was set at d_{max} .

Table III shows the evaluation of the extracted building boundaries by the α -shape algorithm. Comparing these results with those in Table II, it is evident that the proposed boundary extraction algorithm showed better completeness with lower area omission error, although the correctness and area commission error were similar. Note that the worse average results in Table III are highlighted in bold-italic font.

The performance difference can be visualised in a low density point cloud data. So, some examples of boundary extraction from the KN data set are shown in Figures 9. It is

seen that the alpha-shape algorithm misses some true building area (Figure 9a, within orange circle), while the proposed technique did not (Figure 9d). Consequently, the proposed method offered higher completeness and quality for the test scenes. It also showed lower area omission errors.

V. CONCLUSION

A simple but effective solution for building boundary identification from the input point cloud data has been proposed. The initial boundary (in terms of point edges) from the Delaunay triangulation of the input point set is gradually refined to determine the building boundary. The algorithm is capable of determining an appropriate boundary from the input point set of any shapes. It can clearly identify the concavities along the boundary as well as any large concavity inside the shape. It also separately identifies the inside hole boundary, if any. An important and unique property of the proposed technique is that it can extract the boundary of a large point set from the boundaries of two or more of its point subsets. Any inside hole can also be determined from the boundaries of subsets. This is particularly useful for obtaining a building boundary from the boundaries of its planes.

Experimental results have shown that the proposed boundary and line extraction techniques offer better performance in terms of completeness and quality. They can preserve detail along the building boundary, even in low density input data.

ACKNOWLEDGMENT

Dr. Awrangjeb is the recipient of the Discovery Early Career Researcher Award by the Australian Research Council (project number DE120101778). The Vaihingen data set was provided by the German Society for Photogrammetry, Remote Sensing and Geoinformation (DGPF) [8]. The Aitkenvale and Hervey Bay data sets were provided by Ergon Energy in Queensland, Australia. The Knox data set was provided by the Department of Environment and Primary Industries of Victoria, Australia.

REFERENCES

- [1] M. Awrangjeb and C. S. Fraser, "An automatic and threshold-free performance evaluation system for building extraction techniques from airborne LIDAR data," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 7, no. 10, pp. 4184–4198, 2012.
- [2] T. L. Haithcoat, W. Song, and J. D. Hipple, "Building footprint extraction and 3-D reconstruction from LIDAR data," in *IEEE/ISPRS Joint Workshop on Remote Sensing and Data Fusion over Urban Areas*, Rome, Italy, November 2001, pp. 74–78.
- [3] R. A. Jarvis, "Computing the shape hull of points in the plane," in *IEEE Computer Society Conference Pattern Recognition and Image Processing*, 1977, pp. 231–241.
- [4] A. Sampath and J. Shan, "Building boundary tracing and regularization from airborne LIDAR point clouds," *Photogrammetric Engineering & Remote Sensing*, vol. 73, no. 7, pp. 805–812, 2007.
- [5] H. Edelsbrunner, D. G. Kirkpatrick, and R. Seidel, "On the shape of a set of points in the plane," *IEEE Transactions on Information Theory*, vol. IT-29, pp. 551–559, 1983.
- [6] M. Awrangjeb and C. S. Fraser, "Automatic segmentation of raw LIDAR data for extraction of building roofs," *Remote Sensing*, vol. 6, no. 5, pp. 3716–3751, 2014.
- [7] Barista. (2011) The barista software. [Online]. Available: www.baristasoftware.com.au
- [8] M. Cramer, "The DGPF test on digital aerial camera evaluation - overview and test design," *Photogrammetrie Fernerkundung Geoinformation*, vol. 2, pp. 73–82, 2010.