TU Delft

Delft University of Technology

Building outline extraction from als point clouds using medial axis transform descriptors

Widyaningrum, Elyta; Peters, Ravi Y.; Lindenbergh, Roderik C.

DOI
10.1016/j.patcog.2020.107447

Publication date
2020

Document Version
Final published version

Published in
Pattern Recognition

Citation (APA)
Widyaningrum, E., Peters, R. Y., & Lindenbergh, R. C. (2020). Building outline extraction from als point clouds using medial axis transform descriptors. *Pattern Recognition*, *106*, Article 107447. https://doi.org/10.1016/j.patcog.2020.107447

Important note
To cite this publication, please use the final published version (if applicable).
Please check the document version above.

Copyright
Other than for strictly personal use, it is not permitted to download, forward or distribute the text or part of it, without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license such as Creative Commons.

Takedown policy
Please contact us and provide details if you believe this document breaches copyrights.
We will remove access to the work immediately and investigate your claim.

This work is downloaded from Delft University of Technology.
For technical reasons the number of authors shown on this cover page is limited to a maximum of 10.

Pattern Recognition 106 (2020) 107447

Contents lists available at ScienceDirect

Pattern Recognition

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

Building outline extraction from als point clouds using medial axis transform descriptors

Elyta Widyaningrum [a,c,*], Ravi Y. Peters [b], Roderik C. Lindenbergh [a]

[a] Dept. of Geoscience and Remote Sensing, Delft University of Technology, Stevinweg 1, 2628 CN Delft, Netherlands
[b] Dept. of Urbanism, Delft University of Technology, Julianalaan 134, 2628 BL Delft, Netherlands
[c] Centre for Topographic Base Mapping and Toponyms, Geospatial Information Agency, Jalan Raya Jakarta-Bogor KM. 46 16911 Cibinong, Indonesia

ARTICLE INFO

Article history:
Received 7 November 2019
Revised 22 April 2020
Accepted 13 May 2020
Available online 25 May 2020

Keywords:
Skeleton
Point cloud
Building outline
Medial axis transform
Segmentation

ABSTRACT

Automatic building extraction and delineation from airborne LiDAR point cloud data of urban environments is still a challenging task due to the variety and complexity at which buildings appear. The Medial Axis Transform (MAT) is able to describe the geometric shape and topology of an object, but has never been applied for building roof outline extraction. It represents the shape of an object by its centerline, or skeleton structure instead of its boundary. Notably, end points of the MAT in principle coincide with corner points of building outlines. However, the MAT is sensitive to small boundary irregularities, which makes shape detection in airborne point clouds challenging. We propose a robust MAT-based method for detecting building corner points, which are then connected to form a building boundary polygon. First, we approximate the 2D MAT of a set of building edge points acquired by the alpha-shape algorithm to derive a so-called building roof skeleton. We then propose a hierarchical corner-aware segmentation to cluster skeleton points based on their properties which are the so-called separation angle, radius of the maximally inscribe circle, and defining edge point indices. From each segment, a corner point is then estimated by extrapolating the position of the zero radius inscribed circle based on the skeleton point positions within the segment. Our experiment uses point cloud datasets of Makassar, Indonesia and EYE-Amsterdam, The Netherlands. The average positional accuracy of the building outline results for Makassar and EYE-Amsterdam is 65 cm and 70 cm, respectively, which meet one-meter base map accuracy criteria. The results imply that skeletonization is a promising tool to extract relevant geometric information on e.g. building outlines even from far from perfect geographical point cloud data.

© 2020 The Author(s). Published by Elsevier Ltd.
This is an open access article under the CC BY license. (http://creativecommons.org/licenses/by/4.0/)

1. Introduction

Mapping building roof outlines, also called building footprints, is essential for digital base map cartography, planning, surveillance, infrastructure management and sustainable city design. Several urban-related applications such as cadaster maintenance and building taxation require building outlines at a routine basis. Extracting building boundary lines manually is expensive and time consuming, especially in urban scenes. Research on extracting building outlines automatically from high-resolution data remains challenging due to the complexity of roof structures and variations in the design of our urban environment. Up to now, building outlines are typically digitized from multiple aerial images. The use of aerial images is preferred above other data sources as human operators can easily detect buildings on such images. For automation purposes, image disadvantages such as shadows, trees covering building roofs and color variations may increase the extraction error. Moreover, relief displacement may cause problems when using an orthoimage to obtain building boundaries in case of unfavourable image acquisition angles [61]. Airborne Laser Scanner (ALS) point cloud data is an alternative data source. ALS point clouds have been used as a major data source for mapping applications for a few decades [2]. The ability of ALS point clouds to provide many accurate and undistorted 3D points makes it suitable as data source for object extraction. Man-made urban objects (buildings, roads, canals) typically have symmetric shape with straight lines and sharp corners. Such characteristics enable automatic boundary outline extraction from an ALS point cloud. Thus, the use of ALS point clouds for rooftop mapping in combination

* Corresponding author. Geoscience and Remote Sensing, Stevinweg 1, 2628 CN Delft, ZUID HOLLAND, Netherlands.
E-mail addresses: e.widyaningrum@tudelft.nl, elyta.widyaningrum@big.go.id (E. Widyaningrum), r.y.peters@tudelft.nl (R.Y. Peters), r.c.lindenbergh@tudelft.nl (R.C. Lindenbergh).

https://doi.org/10.1016/j.patcog.2020.107447
0031-3203/© 2020 The Author(s). Published by Elsevier Ltd. This is an open access article under the CC BY license. (http://creativecommons.org/licenses/by/4.0/)

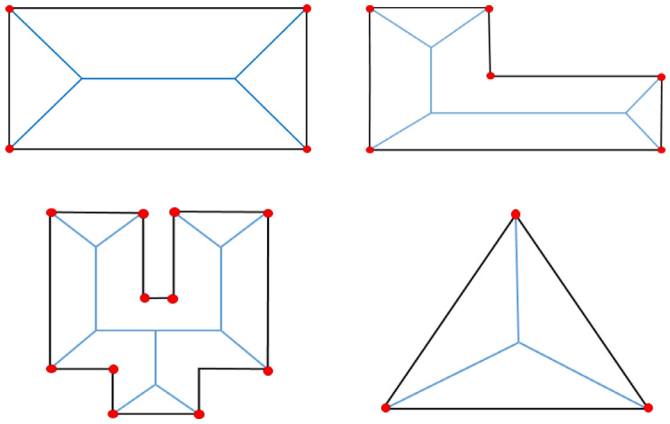


Fig. 1. The MAT skeleton (blue lines) intuitively detects corners (red points) located at the intersection of the skeleton and the object boundary (black lines).

with an efficient algorithm is expected to provide better building outlines.

Medial axis transform (MAT), is a powerful shape extraction technique that provides a compact geometrical representation while preserving topological properties of the input shape [1,4]. The MAT was introduced by Blum [5] to describe biological shapes. Since then, it has been used for applications in image processing and computer vision. However, MAT has a fundamental drawback, which is its instability to small perturbations of the input shape, which then may disturb the topology of the MAT branches [3,9]. Moreover, wider deployment of MAT to extract shapes analysis from surveying quality data with its associated problems, is still challenging [4].

In principle, the MAT can be implemented for urban mapping purposes, particularly to extract building shapes by detecting its corners. As illustrated in Fig. 1, corner points (red) are detected when the tip of a skeleton branch (blue line) touches the building boundaries (black). However, generating a MAT skeleton from point clouds is a challenging problem as such data contain fuzzy borders, in particular, when data is missing [25]. This makes the application of MAT for airborne point clouds difficult.

Corners are important local features and knowledge on their location can minimize further data processing without losing specific features of the original object shape [35,36]. Given an airborne point cloud of an urban area, we propose a method for extracting building outlines automatically by detecting accurate roof corner points based on MAT descriptors.

2. Related work

Work on the development of building outline extraction from various remote sensing data has been intensified in parallel with the increased interest in GIS (Geographic Information System) digital map products [49]. Combining different data sources to extract building outlines is believed to increase the detection rate and accuracy compared to using a single data source as more features can be used. Nevertheless, fusion of data of different sensor types is not a trivial task as fusion is hampered by dissimilar resolution, alignment issues or mismatches in feature information caused by sensor characteristics or differences in viewpoint during acquisition [60]. Manno-Kovacs and Sziranyi [54] proposed an Orientation Selective Building Detection method to detect buildings from a combination of aerial and high-resolution satellite images. They apply active contouring for obtaining smooth and accurate building outlines. However, inhomogeneous buildings are sometimes only partially detected and any object connected to the building (e.g. trees) can result in false positive detections. Zhao et al. [55], Awrangjeb [56], and Li et al. [57] combined LiDAR point clouds and aerial images to detect buildings and obtain smooth building outlines by regularization and mathematical morphology. Building outline errors occured due to failures to determine the building principal directions during regularization [55,56] or line redundancies after simplification [57] especially, in case of low point density. Our previous study [37] proposed an extended Hough transform method using ordered lists of points to detect building boundary segments from airborne point cloud data. Hierarchical filtering is applied to remove spurious lines. That method outperformed existing state of the art methods in terms of correctness and quality metrics on benchmark dataset. However, the method is likely to introduce false corners, especially for buildings of complex shape, as spurious lines may still exist in the final step.

Various definitions of MAT or skeleton found in literature correspond to different methods for computing the MAT leading to different results with different properties. In general, MAT algorithms typically focus on deriving the geometric location of the centerline or medial axis of a surface, so-called skeletonization [11]. Up to now, numerous skeletonization methods and their application for 2D and 3D object description are available in literature. The existing skeletonization methods are often categorized into four main approaches:

- morphological thinning-based methods that was first applied for discrete binary images [17], and improved by Huang et al. [22] and other related notions on the development of 3D thinning algorithm [23,48].
- geometry-based methods using medial axis transformation for planar shape [18] including Voronoi diagrams [19] and Delaunay triangulation [47].
- distance-based functions such as skeleton generation and centerline by the distance transform [40], skeletonization by discrete Euclidean distance maps [41], and Euclidean skeleton based on connectivity criterion [42].
- general-field functions which are generated by functions rather than use distance function, for example by replacing the nonlinear distance with a linear transform [16], using Newtonian potential model to replace the distance function [43], and using Electrostatic Field Theory (EFT) function to generate potential distribution inside the object [44].

Reviews on skeletonization methods and its applications have been discussed by Saha et al. [12,13], Pavlidis [21], and Amenta et al. [24]. As many skeletonization algorithms were designed more for image analysis, we limit the scope of our study on skeletonization for point cloud data.

Several works on MAT using geographical data for various purposes have been conducted. Haunert and Sester [30] applied straight skeleton extraction to derive linear representations of polygons and road centerlines from a cadastral dataset. Yirci et al. [29] extracted detailed pedestrian networks by generating a centerline using two skeleton operators (straight skeleton by parallel thinning and medial axis by Voronoi diagram). Methods for river centerline extraction based on Delaunay triangulations remain challenging for certain complex situations e.g. a scenario with a skeleton branching in different directions [33,34]. Broersen et al. [26] used the 2D skeleton of a Voronoi diagram and the 3D skeleton of a shrinking ball for identifying watercourses and deriving its centerlines from classified aerial point clouds. Widyaningrum and Lindenbergh [38] extract the road network of an urban area from a colored point cloud using parallel thinning skeletonization [39]. However, a further generalization step maintaining the road topological order is required for smoothing jaggy skeletal lines yielded by the parallel thinning algorithm.

Ma et al. [6] estimate a 3D medial axis point using a shrinking ball approach based on nearest neighbors and normals from

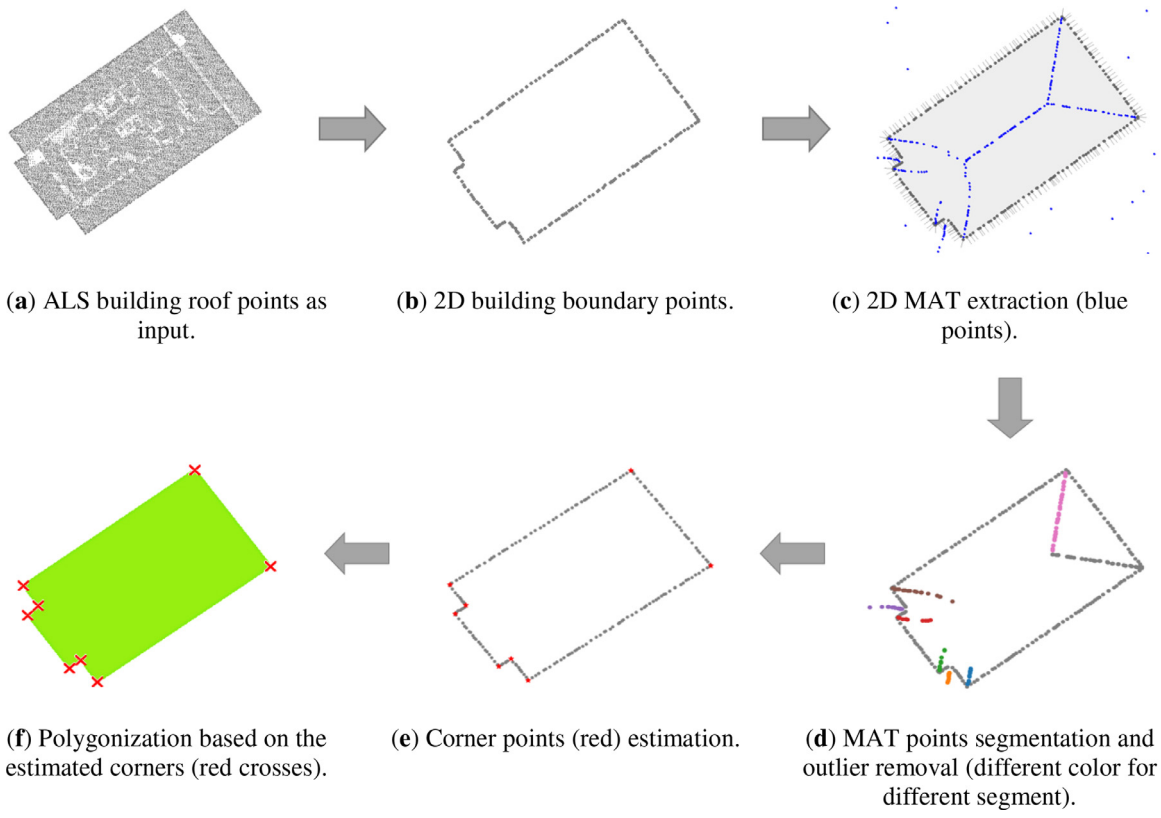


Fig. 2. Proposed methodological workflow for extracting the building outlines.

a given set of points. Their work is claimed as a faster and simple method that approximates the 3D MAT based on maximally inscribed balls tangent to two surface points whose center is positioned on a normal line. The ball centers are the medial axis points. The shrinking ball algorithm is not only accurate and computationally efficient but is also considered as the most simple and fast existing surface-skeletonization method [4]. This method is considered as suitable for the geographical case as it is point based, simple, fast, and scalable [3]. However, to use this method, fine sampling is required to directly obtain a high quality skeleton. The fact that the shrinking ball approach can only result in unstructured skeleton points while disregarding the topology of the skeleton branches makes this algorithm not directly applicable for applications involving surveying data. For use in practical applications, the MAT consisting of medial points and corresponding maximally inscribed circles (in 2D) or balls (in 3D) needs further processing [10,14,15].

3. Methodology

Our research focuses on the adaptation of MAT for extracting building outlines from noisy point cloud data required for mapping and spatial modeling purposes. We extend the work on the iterative shrinking ball algorithm and develop a strategy to exploit skeleton features to accomplish the goal of accurate building outline extraction. A new approach for skeletal point segmentation is also proposed in this study. The proposed method achieves state-of-the-art in handling noisy surface boundaries and reconstructing the building outlines. It requires minimal human interaction by optimizing the use of skeleton-based features. Specifically, the contributions of our work are as follows:

- We integrate skeleton-derived features and global features to perform robust skeletal points (MAT) segmentation handling varying point density and noise level.
- We combine ordered surface point indices and skeletal-derived features to detect corner points.
- We introduce the use of skeletal-derived features to estimate building corner positions accurately.

Overall, our method overcomes some traditional pitfalls of using MAT techniques in case of noisy input.

As this study requires using the MAT in 2D space, we adapt the 2D shrinking circle algorithm by Ma et al. [5]. The general workflow of our proposed method for automatically extracting building roof outlines consists of four main steps (see Fig. 2). First, building boundary points are extracted by an alpha-shape algorithm [45]. Next, the boundary points are transformed into its 2D MAT or skeleton points using the 2D shrinking circle algorithm. Third, we then apply our MAT segmentation to segment the MAT points by exploiting their geometric attributes. The segments are then used to detect corner points. Fourth, polygonization is carried out to form a 2D closed polyline based on the detected corner points.

This study uses the extended shrinking circle approach that implements the denoising heuristic as proposed by Peters [3]. We define the skeleton of an object surface $\boldsymbol{S}$ as a set of center points c of maximally inscribed circles $\boldsymbol{B}(c, \rho)$ in $\boldsymbol{S}$ (see Fig. 3) where ρ denotes the radius of such circle.

The 2D skeleton points are also called medial axis points. By associating the circle radius ρ function to the set of medial axis points, we obtain the so-called Medial Axis Transform (MAT). As shown in Fig. 3, the medial axis points (red points) form the MAT skeleton (blue lines) of a rectangular object $\boldsymbol{S}$. Each maximally inscribed circle (in grey) touches at least two points of the boundary of $\boldsymbol{S}$ (black outline). Center points of any circle that is not maximal

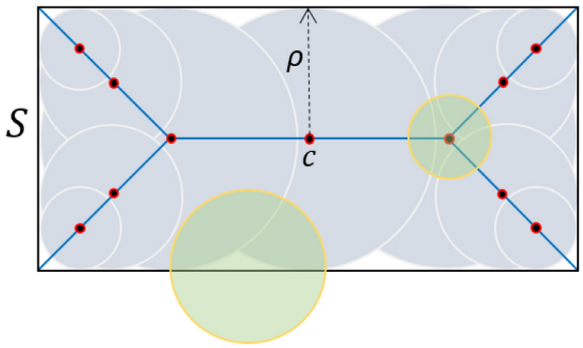


Fig. 3. The skeleton (blue line) of a rectangular shape with its corresponding inscribed circle (grey) and medial axis point *c* (red point).

or not inscribed in ***S*** (green circles) are dismissed and not considered as medial axis points.

To provide a clear and coherent narrative, further details on each methodological step are provided in the following: Section III.1 describes the alpha-shape algorithm. Section III. 2 and III. 3 provide necessary details on the shrinking circle method and skeletal points extraction, respectively. Skeletal point segmentation is described in Section III. 4. The last step, corner point estimation and building outline generation, is explained in Section III. 5. Evaluation methods for building outline extraction used in this research are discussed in Section III. 6.

3.1. Alpha-shape

Given the segmented building points, creating building outlines starts with boundary point selection by the alpha-shape algorithm as introduced by Edelsbrunner [45]. An alpha-shape is well known for its capability to preserve small shape details of a finite point set at a required level of detail. The 2D alpha-shape is constructed based on the 2D Delaunay triangulation of the input points. The method identifies boundary points that are defined in terms of a parameter $\alpha \geq 0$, which controls the level of detail of the boundary shape. Given a set S of points on a plane and a value of α, the algorithm works as follows:

1. Compute the Delaunay triangulation DT(S) of S. All edges in DT(S) are candidate for the alpha-shape S_α.
2. For all edges e of DT(S) with end points p and q, say:
   a. Find two circles B_{pq1} and B_{pq2} of radius α with center $c_{pq}(1)$ and $c_{pq}(2)$ containing end points p and point q of the same edge e. The circles are defined in terms of the below circle centers:

$$c_{pq}(1,2) = \left(\frac{x_p + x_q}{2} \pm \sqrt{\alpha^2 - \left(\frac{\|e\|}{2}\right)^2} \left(\frac{y_p - y_q}{2}\right), \frac{y_p + y_q}{2} \pm \sqrt{\alpha^2 - \left(\frac{\|e\|}{2}\right)^2} \left(\frac{x_p - x_q}{2}\right) \right) \quad (1)$$

   Where $\|e\|$ is the length of the edge between end points p and q.
   b. If at least one of the circles contains no points from S in its interior, e is a valid boundary edge (Fig. 4.a), otherwise the edge is removed (Fig. 4.b).
3. The union of all valid boundary edges forms the alpha-shape S_α (Fig. 4.c)

The value of α is a real number with $0 \leq \alpha \leq \infty$. As α approaches 0, the shape may shrink, develop holes and may become disconnected. In the extreme case, the value of $\alpha = 0$ results in the data points itself. When α increases towards infinity, the alpha shape approaches the *convex* hull of the set S of points. In case of geospatial point clouds, the point density is often varying, and is depending on sensor characteristics and measurement geometry and an appropriate value of α should be chosen accordingly. To identify boundary points of unordered building roof points in our study areas, we decided empirically for an α-value between 0.3 and 0.5

3.2. The shrinking circle principles

Given are a set of noisy edge points V on a surface ***S*** with corresponding normal vectors N. The MAT points are defined as the set of centers c and corresponding radius ρ of maximally inscribed circles ***B***(c, ρ) in ***S*** that are bi-tangent to the boundary ***S***. The circle ***B*** and corresponding circle center c are denoted as medial circle and medial axis point, respectively.

The basic principles of the shrinking circle method (see Fig. 5) are as follows:

1. A medial circle touches the surface in at least two points (p, q) where $p, q \in$ ***S***.
2. Following the line defined by normal vector N_p of edge point p, the radius ρ of a circle $\boldsymbol{B}_p$ decreases iteratively until $\boldsymbol{B}_p$ touches ***S*** at q, where $q \neq p$ and the circle center c is on the line through N_p. Iteration stops if the maximal $\boldsymbol{B}_p$ circle is found.
3. A medial circle is a maximal empty circle, which means it contains no surface points.

3.3. Skeletal points extraction

To obtain the MAT of surface ***S***, medial axis points $c(p)$ are computed. Hence, the maximal inscribed medial circle ***B*** for all sample points p in ***S*** is computed by the following steps:

1. An initial circle $\boldsymbol{B}_{init}$ of p is defined based on an initial radius ρ_{init}. The ρ_{init} value is set sufficiently large e.g. equal to the largest distance between two input points.
2. Given ρ_p^k, where $k = \{1, 2, .., i\}$ denotes the k-th iteration step, the circle center $\boldsymbol{c_p^k}$ is given by:

$$c_p^k = \boldsymbol{p} - \boldsymbol{N_p}\, \rho_p^k \quad (2)$$

3. Find the surface point $q_p^k \in \boldsymbol{S}$ closest to c_p^k such that $q_p^k \neq p$.
4. Test for circle maximality for the circle defined by q_p^k and p:
   a. If the distance from c_p^k to q_p^k equals the radius of the circle ρ_p^k, the circle $\boldsymbol{B_p^k}$ is maximal and c_p^k is a medial axis point.
   b. Otherwise, compute the radius of the next shrinked circle ρ_p^{k+1} using the following equations:

$$\rho_p^{k+1} = \frac{\boldsymbol{d}(\boldsymbol{p}, \boldsymbol{q}^{k+1})}{2\boldsymbol{cos}\theta_p^{k+1}} \quad (3)$$

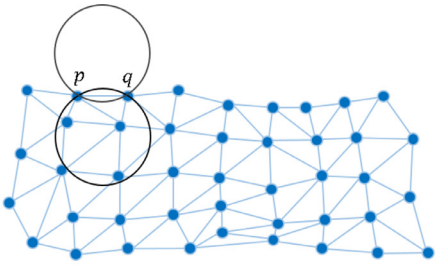
   Where:

$$cos\theta_p^{k+1} = \frac{\boldsymbol{N}\left(\boldsymbol{p} - \boldsymbol{q}^{k+1}\right)}{\boldsymbol{d}(\boldsymbol{p}, \boldsymbol{q}^{k+1})} \quad (4)$$

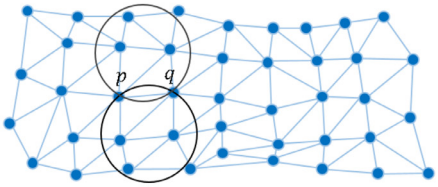
$$d(p, q^{k+1}) = \frac{\left|\boldsymbol{p} - \boldsymbol{q}^{k+1}\right|}{2} \quad (5)$$

The iteration will stop when the medial axis point as described in step 5.a. is found. Fig. 5.a. shows consecutive shrinking of a circle touching ***S*** at point p, which results in a medial circle $\boldsymbol{B_p^i}$ and a medial axis point c_p^i in the last iteration.

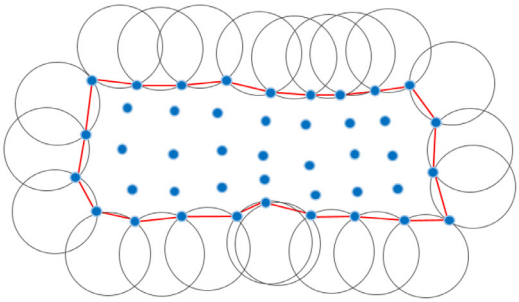
Given a defined inside and outside of surface ***S***, the MAT consists of two components: one part inside surface ***S***(N_p), consisting



(**a**) Delaunay edge connecting points p and q is a boundary edge as one of the circles, here the top circle, is empty.

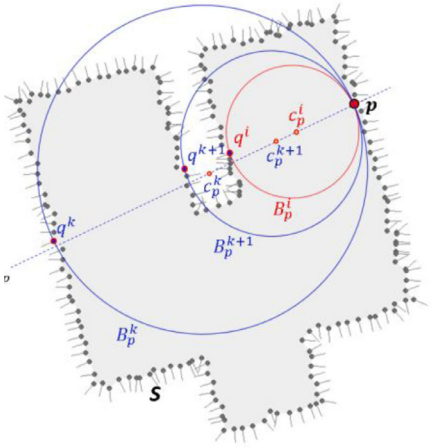


(**b**) Delaunay edge connecting points p and q is not a boundary edge as both circles are not empty.

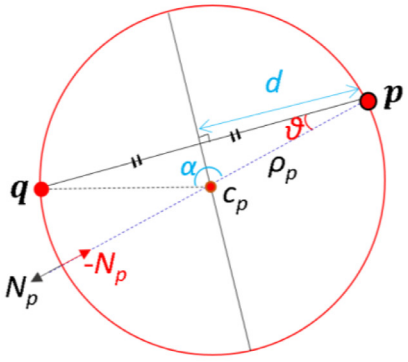


(**c**) Complete boundary polygon (red line) of the point set as indicated by empty circles of radius α (black circles).

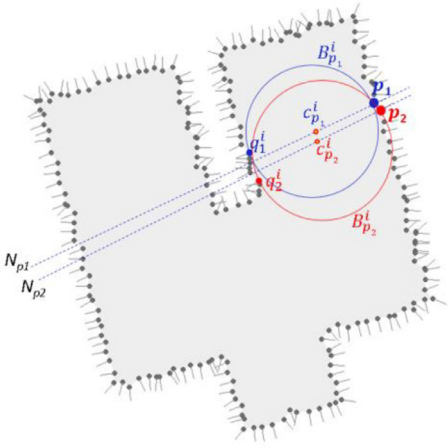
Fig. 4. The alpha-shape criteria.



(**a**) Shrinking circle iteration of point p.



(**b**) Circle center c_p is the intersection of N_p and perpendicular line of bisector $\overline{pq}$.



(**c**) Medial circles of two consecutive points p_1 and p_2 results in two medial axis points c_{p1}^i and c_{p2}^i.

Fig. 5. Basic principles of the shrinking circle algorithm applied on noisy building edges.

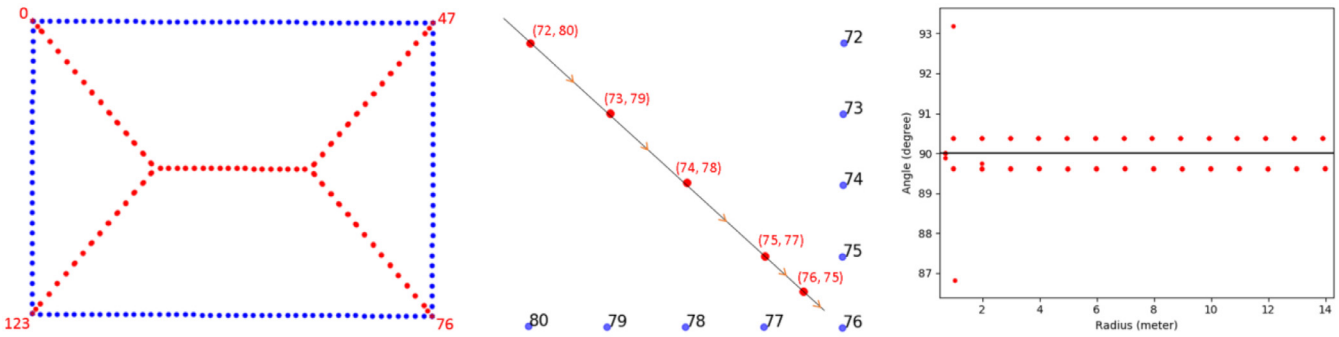
of the so-called inner medial axis points, and another one outside surface $\boldsymbol{S}(-N_p)$, the outer medial axis points.

For each $p \in \boldsymbol{S}$, the corresponding inner and outer MAT points are computed by iterating steps 2 to 4. The inward normal N_p is used for the inner MAT calculation, while the outward normal–N_p is used for the outer MAT calculation. Fig. 5.b. shows the geometry for calculating the medial axis point c_p and the direction of the normal vector N_p for the inner circle (black arrow) and outer circle (red arrow).

Noise handling is an essential step to overcome the sensitivity of MAT to noisy boundaries. In case a small bump or noise exist on the input surface, a circle may get shrunk too much which then likely results in undesireable medial axis points. Such overly shrinked circle, typically has a small separation angle α. The separation angle α (see Fig. 5.b) is the angle between line p - $\boldsymbol{c_p}$ (the line connecting point p and medial axis point $\boldsymbol{c_p}$) and line q - $\boldsymbol{c_p}$ (the line connecting point q and medial axis point $\boldsymbol{c_p}$).

$$cos\alpha = \frac{\overrightarrow{c_p p} \cdot \overrightarrow{c_p q}}{\left|\overrightarrow{c_p p}\right| \cdot \left|\overrightarrow{c_p q}\right|} \tag{6}$$

The denoising heuristics technique presented by Peters [2018] is used to select the so-called good circle that is often computed



(**a**) MAT points (red) heading to the same turning point of the edge are considered as one segment. Turning points of this shape are: 0, 47, 76, and 123.

(**b**) Median of the corresponding (p, q) of MAT point (red) is the same as the turning point index (76).

(**c**) Distribution of MAT points (red) angle that are close to 90°.

Fig. 6. Characteristics of skeletal points used for segmentation in case of uniform edge point spacing.

during the above described shrinking approach. The good circle is defined as the last circle in the shrinking approach with a separation angle α_k bigger than the separation angle threshold α_{min}.

After this step, every medial axis point $\boldsymbol{c_p}$comes with a number of attributes. Attributes of each MAT point $\boldsymbol{m_p}$ are medial axis point $c_p$coordinate, radius ρ, separation angle α, indices of surface point p and q, and normal vector N_p or -N_p. Theoretically, using these MAT attributes, the geometry of $\boldsymbol{S}$ can be reconstructed completely.

3.4. MAT point segmentation

MAT attributes provide rich information that can be used to group MAT points into different medial segments or branches. In our case, segmenting the points into different branches is used for detecting the corner points. One issue when obtaining a skeleton from a point cloud with the shrinking circle algorithm is that the point cloud provides an unstructured point sampling of $\boldsymbol{S}$. Also for the MAT points resulting from steps 1 to 4, adjacency relations are initially not known. For further application of the MAT, two useful observations to identify MAT point connectivity are as follows:

- MAT points heading towards the same turning point or corner are considered as one segment. Fine sampled surface points of a square shape $\boldsymbol{S}$ in Fig. 6.a result in fine MAT points in which some of the MAT points gradually approach a specific turning point. In this sense, each MAT point created from a maximally inscribed circle, touching at surface point p and q appoint to a turning point that is equally located between surface point p and q. As illustrated in Fig. 6.b, the median value of two surface points p and q (in red text) is similar to the corner's index (76).
- MAT points are expected to have a separation angle α close to 90° As shown in Fig. 6.c, MAT points of a rectangular shape have separation angles that are distributed around 90°

In practice, surface points are not perfectly distributed and noise-free and are not as regular as shown in Fig. 6.a. Small perturbations on the surface boundary create so-called skeletal noise [7,8]. When detecting shape corners, skeletal noise may induce false segments, which then results in false corners.

Our segmentation criterion relies on the proximity of the turning point location of the boundary point. Intuitively, MAT points that lie close to each other and have similar features values are grouped together. Moreover, we expect that the radius ρ will gradually change along a segment branch.

Based on aforementioned observations, we use three global thresholds and four MAT-derived features for segmenting the MAT points. The global thresholds are not related to MAT and are defined to increase the segmentation accuracy. The global thresholds are:

G.1. A buffer distance *bf* from the object surface $\boldsymbol{S}$. Only a MAT point located within the specified buffer will be considered for segmentation. This threshold is used to exclude unusable outer MAT points resulting from the maximal circle of two edge points with outward normal. These MAT points are typical noise located far from the surface points.

G.2. Minimum number of points for each segment *minPts.* Any segment having less points than the given *minPts* is considered as segment noise.

G.3. Point index interval Δpt sets the minimum distance between two candidate corner points, as expressed below:

$$\Delta pt \geq \frac{l}{r} - 2 \tag{7}$$

In Eq. (7), l is the minimum required edge length and r is the point cloud interval. The point index interval Δpt criterion is designed to avoid having false or extra corners at a certain minimum determined edge length in case of short and noisy boundaries. For example, given a set of points with 0.5 m point cloud interval r, we require to extract building edges of minimum length $l = 2.5$ m, thus, Δpt is set to 3. Imagine that point 13 in Fig. 7 has the same medial properties as point 11, 16, and 20. Point 13 will not be considered as a corner point as it has less than 3 point difference to point 11.

The customized features derived from the MAT attributes, or MAT-derived features are described as follows:

F.1. A MAT point $\boldsymbol{m_p}$ having a separation angle α close to 90° is considered for segmentation. Here 'close' is specified by the separation angle difference threshold, $\partial\alpha$. MAT points outside the given $\partial\alpha$ threshold are considered as skeletal noise. This means, a MAT point $\boldsymbol{m_p}$ will be considered for segmentation if it has a separation angle $\alpha_{\boldsymbol{p}}$ between $90° + \partial\alpha$ and $90° - \partial\alpha$.

$$90° - \partial\alpha \leq \alpha_{\boldsymbol{p}} \leq 90° + \partial\alpha \tag{8}$$

F.2. Each edge point $p \in \boldsymbol{S}$ is assigned a unique index. For the corner-aware segmentation, MAT points $\boldsymbol{m_p}$ of similar characteristics are expected to belong to the same cluster. This

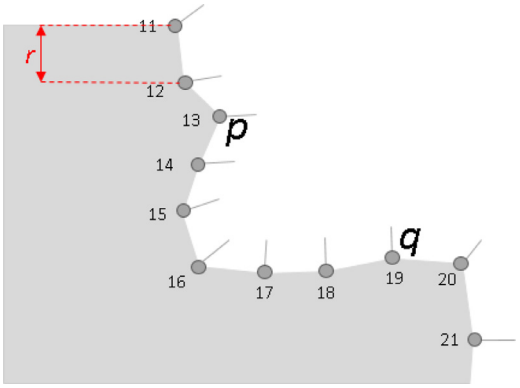


Fig. 7. Noisy point edge may indicate a false corner.

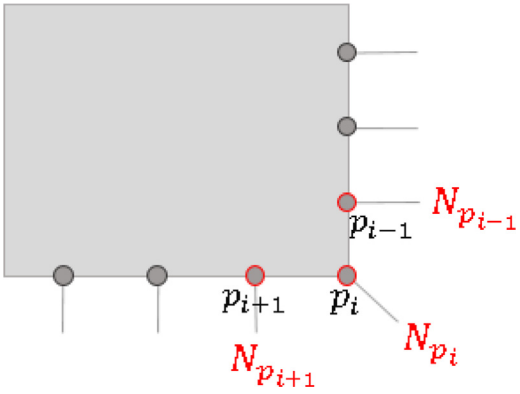


Fig. 8. Normal vectors (grey lines) of edge points (black points) where p_i is the turning point.

is assessed by considering the point indices of the surface points of ***S***. Assume $p, q \in \boldsymbol{S}$. Let p_{idx} and q_{idx} denote the point indices of p and q, respectively. The median value med_{pq} is obtained by Eq. (9).

$$med_{pq} = p_{idx} + \frac{q_{idx} - p_{idx}}{2} \tag{9}$$

For example, p_{idx}=13 and q_{idx}=19 in Fig. 7 results in med_{pq}= 16. Different MAT points with similar med_{pq} value likely belong to the same MAT segment.

F.3. The normal angle differences δN between the normal of point p_i and the normal of the previous point p_{i-1} and the normal of the next point p_{i+1} defined by $(|Np - Np_{i-1}|)$ and $(|Np - Np_{i+1}|)$, respectively.
The normal angle differences δN between 2 consecutive edge points is used to determine candidate corners K_p for $p \in \boldsymbol{S}$. Detection of candidate corners K_p is used to obtain first estimate of the number of building corners which is later used to remove false segments in case of noisy edges. At noisy edges, segments with small median difference may be formed which later may result in false corners. Consider two adjacent normal vectors N_{p_i} and $N_{p_{i-1}}$ of point p and p_{i-1} respectively, compare Fig. 8. The angle δN between the two normal vectors N_{p_i} and $N_{p_{i-1}}$ is obtained via Eq. (10).

$$\cos\left(\delta N_{p_i p_{i-1}}\right) = \frac{N_{p_i} \cdot N_{p_{i-1}}}{|N_{p_i}| . \left|N_{p_{i-1}}\right|} \tag{10}$$

A surface point p initiates a candidate corner point K_p if the angular differences δN to its two adjacent points (p_{i-1} and p_{i+1}) are above the given angle threshold δN. That is expressed in Eq. (11).

$$\left(\delta N_{p_i p_{i-1}}, \delta N_{p_i p_{i+1}}\right) \geq \delta N. \tag{11}$$

For example: for $\delta N = 20^\circ$, if $\delta N_{p_i p_{i-1}} \geq 20^\circ$, and $\delta N_{p_i p_{i+1}} \geq 20^\circ$, then the surface point p is a candidate corner.

F.4. The maximum median index difference threshold ∂K_p. This feature is used to avoid false corners and additional segments caused by perturbation or noise on the surface particularly near to the corners.

Given a set of MAT points $M = \{(c_p, \rho, \alpha, p, q, N)\}$ with 6 features ρ, α, p, q, N per medial axis point c_p, the MAT point segmentation works as follows:

1. Select MAT points $\boldsymbol{m_p}$ located within a certain buffer distance of bf from the object surface ***S*** using global threshold G.1. For most of our buildings, a buffer distance $bf = 35\ m$ is sufficient.
2. Select only MAT points $\boldsymbol{m_p}$ having an acceptable separation angle α_p as specified by the MAT-derived feature F.1. This step eliminates skeletal noise that typically has a separation angle value away from 90° In our case, a separation angle difference $\partial\alpha = 20^\circ$ is sufficient for the medial segmentation.
3. Compute the median value $med_{pq}$of the filtered MAT points from step 2 using the MAT-derived feature F.2.
4. Identify all possible candidate corners K_r and put them in a list $K_1, K_2 \ldots, K_i$. Candidate corner K_r is added to the list if it satisfies all the MAT-derived criteria specified in F.3, F.4, and global feature G.3. We use threshold values ∂K_r= 3, δN = 15°and Δpt= 2, respectively.
5. Given a candidate corner K_r from step 3, the algorithm searches for MAT points with median value med_{pq}similar to r_{idx}.
6. If $|med_{pq} - r_{idx}| \leq \Delta pt$, then MAT point $\boldsymbol{m_p}$ is assigned to medial segment ***Mseg***(r).
7. Any medial segment ***Mseg*** having less member points than *minPts*, as defined in G.2, will be removed. This step will eliminate false segments that may be formed in case of flaws on the edges.

Medial segments are used next to estimate real corners where one medial segment corresponds to one corner.

3.5. Corner point estimation

Instead of appointing edge points as corners, we rather estimate the position of corners based on the medial axis point positions and their corresponding radius. The radius ρ of the maximally inscribed circles of MAT points will gradually decrease towards corners. Each medial segment ideally contains a set of MAT points with gradually decreasing radii. The location where the radius will become zero ($c_{\rho=0}$), typically identifies the location of the corner point corresponding to a medial segment is estimated as follows. Fig. 9 shows how the radius ρ of the MAT point depends linearly on the x coordinate. Therefore, a line is fitted by PCA (Principle Component Analysis) through the (x, ρ) points of the segment at hand. The x- coordinate corresponding to zero radius ($\rho = 0$) of the fitted line L_r (as indicated by the blue line in Fig. 9) is reported as the x coordinate of the corner point. The y-coordinate of the corner point is obtained in a similar way.

In case of an edge with heavy noise, false segment may remain. A spatial filtering step is necessary as final filter to remove any spurious estimated corner c. This spatial filtering step preserves any corner point that is within a specified radius from the surface points $p \in \boldsymbol{S}$. In our case, we use 1 m as we require building outline result to have positional accuracy at least 1 m. As final step, a closed building polygon is obtained by connecting all corner points consecutively referring to the point indices.

3.6. Building outline evaluation metrics

Two different evaluation metrics are applied for evaluating the performance of the proposed workflow in fulfilling the required

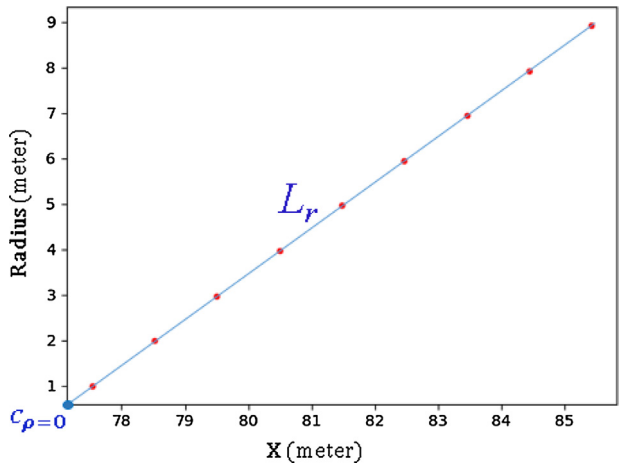


Fig. 9. Estimating the x-coordinate of the corner (blue point) by predicting where radius $\rho_p^k = 0$ on L_r.

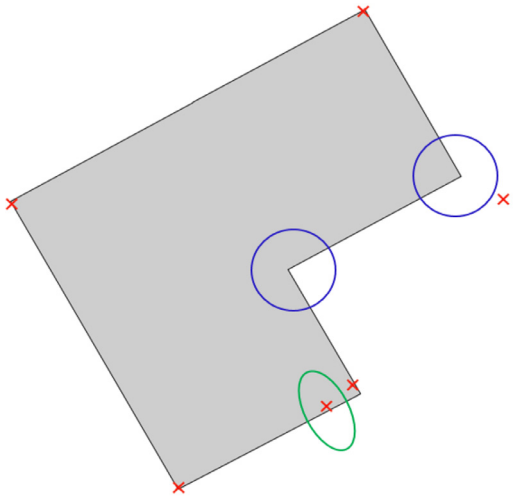


Fig. 10. Illustration of building corner detection accuracy based on the number of correctly detected corners within 1-meter radius from the corner reference. Red crosses indicate building corner results. In blue, two 1-meter circles around an undetected reference corner are indicated. The green ellipse indicates a corner that does not exist in the reference.

building outline specification: corner geometric accuracy and corner detection accuracy. To evaluate the geometric accuracy, we compared the position of corners coordinates of the building outline results to the reference [46]. Positional accuracy, also known as geometric position accuracy or location accuracy, is used as main indicator to measure how well the building polygons are positioned with respect to its true position within an absolute georeferenced system. We use the RMSE (Root Mean Square Error) to measure the average of the squared differences between building corner positions (X and Y coordinate) in the reference and in the result. The RMSE of a complete building is calculated for all detected building corners with respect to the position of corresponding reference corners.

$$RMSEx = \frac{\sqrt{\sum (X_{res} - X_{ref})^2}}{n} \tag{12}$$

$$RMSEy = \frac{\sqrt{\sum (Y_{res} - Y_{ref})^2}}{n} \tag{13}$$

$$RMSEr = \sqrt{RMSEx^2 + RMSEy^2} \tag{14}$$

Where:

X_{res}, Y_{res} = Coordinates of resulting corner points
X_{ref}, Y_{ref} = Coordinates of corner points in the ground truth
n = total number of corner points

Due to the complexity of some building, not all corners may be detected completely. Therefore, we evaluate the corner detection accuracy by means of three retrieval measurements: recall, precision and F1-score [59]. Precision is used to measure the exactness or fidelity, whereas recall is used to measure the completeness. The F1-score is the weighted mean of precision and recall.

$$Precision = \frac{TP}{TP + FP} \tag{15}$$

$$Recall = \frac{TP}{TP + FN} \tag{16}$$

$$F1 - score = \frac{2 \,.\, Precision \,.\, Recall}{Precision + Recall} \tag{17}$$

For this purpose, a corner point is considered a True Positive (TP) if it is located within 1-meter radius from the corresponding corner reference, while any undetected corner including corners with an offset of more than 1 m from the corresponding reference corner is considered a False Negative (FN). Any corner in the result that does not exist in the reference is considered as False Positive (FP).

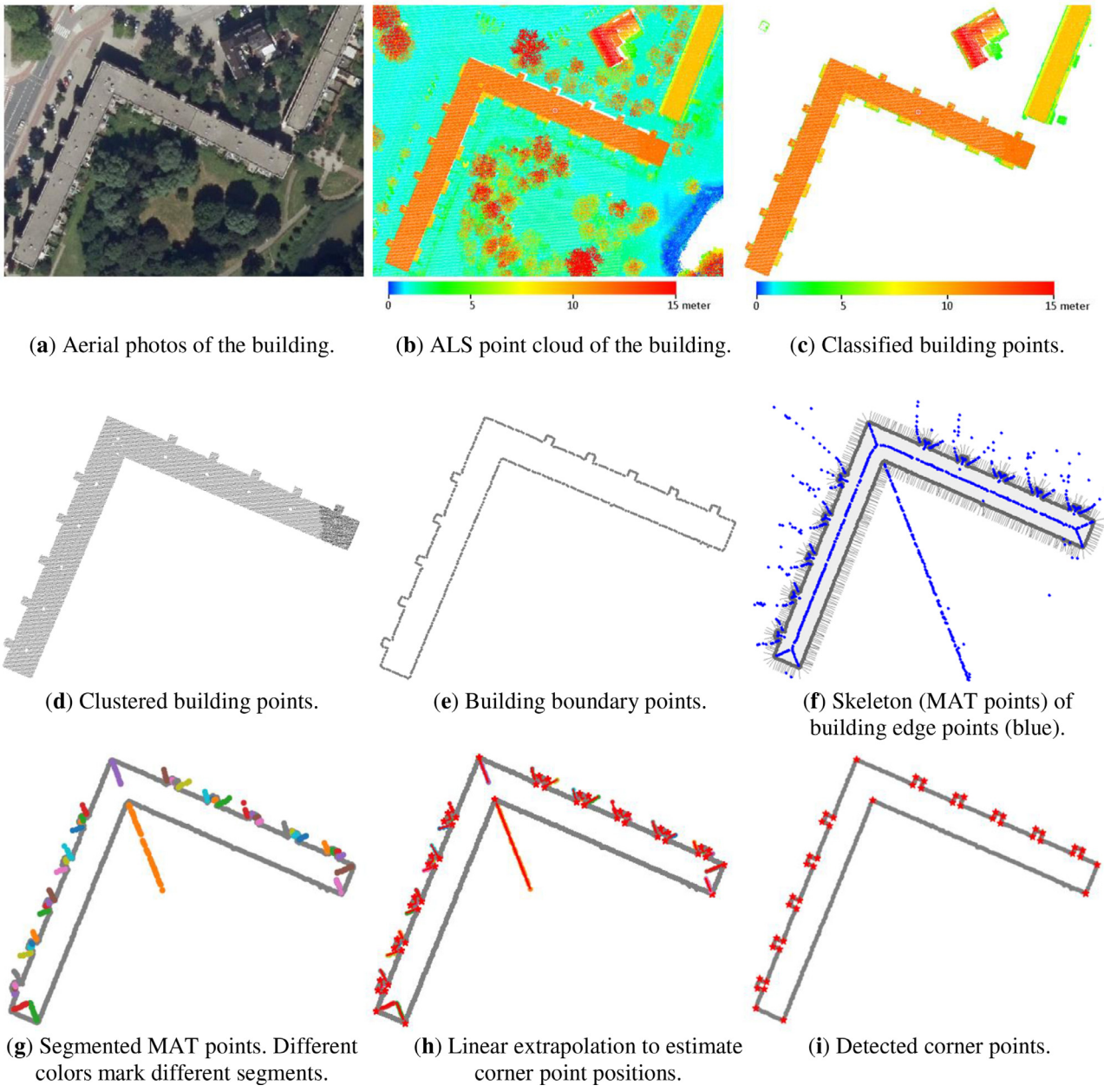
In the building polygon in Fig. 10., the number of correct corners (TP) is 4, while the number of false corners FP (inside the green ellipse) is 1, and the number FN of undetected corners or corners with an offset of more than one meter (as indicated by the blue circles) is 2. This configuration gives Precision = 0.8, Recall = 0.67, and its F1-score is 0.73.

Fig. 11 summarizes the proposed method to detect building corners from a given cluster of building points (Fig. 11.d). The parameter thresholds, as discussed above, are set empirically depending on the point density and the required specification. Additional pre-processing is necessary in case clustered building points are not available. In this case, an initial classification and/or semantic segmentation processing step is required. We use a classified ALS point cloud whose points are labelled according to their object class (building, ground, and unclassified). To group points belonging to one building, points are clustered by applying the DBSCAN algorithm [50]. As a result, different buildings will have different cluster number and theirs points are labelled according to the corresponding cluster number. Once the clustered building points (as presented in Fig. 11.d) are available, boundary points need to be extracted (Fig. 11.e), for which we use the alpha-shape algorithm. The resulting building boundary points are then used as input for the MAT shrinking circle algorithm (blue points Fig. 11.f). MAT points filtering and segmentation is applied based on their separation angle and median index value (Fig. 11.g). Each medial segment generates a different corner candidate. Positions of the corners are extrapolated linearly using PCA (Fig. 11.h).

4. Results and discussions

4.1. Experiments of the study areas

For the experiment, we use three study areas with different landscape characteristics and airborne LiDAR point cloud specifications. The first dataset represents a sub-urban area of the city of Makassar, Indonesia. The point cloud data was captured in 2012 by a Leica ALS70 instrument and has 7–11 ppm (point per meter) point density.



(**a**) Aerial photos of the building. (**b**) ALS point cloud of the building. (**c**) Classified building points.

(**d**) Clustered building points. (**e**) Building boundary points. (**f**) Skeleton (MAT points) of building edge points (blue).

(**g**) Segmented MAT points. Different colors mark different segments. (**h**) Linear extrapolation to estimate corner point positions. (**i**) Detected corner points.

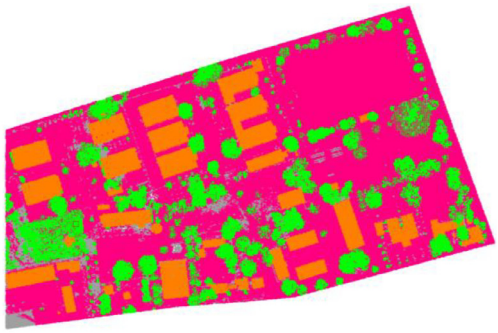
Fig. 11. Overview of the proposed corner detection method using MAT descriptors.

The second dataset is a Dutch national AHN3 point cloud [31] sampling the area of EYE-Amsterdam, the Netherlands. The AHN3 data has a point density of at least 10 ppm and was acquired in 2014. Most buildings in this area have a public or business function. This test set is selected as many of the buildings in this area are considered to have a high complexity in terms of shape and size.

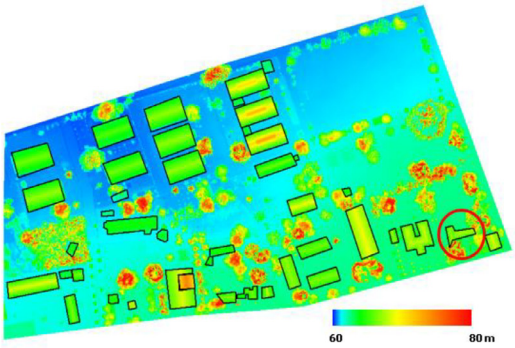
Building points of the Makassar dataset, as presented in Fig. 12.a. in orange, were classified using LAStools [27]. For EYE-Amsterdam, we used the provided building classification of the AHN3 dataset (shown as orange points in Fig. 12). The alpha-shape algorithm [28] is then applied to derive the outline of each building. We do not discuss the details of the pre-processing steps further as it combines well-known methods in the field of GIS and remote sensing.

For the Makassar area, the topographic base map scale 1:10.000 is used as ground truth data. The topographic base map is generated from manual 3D delineation from stereo-images with the same acquisition time as the Makassar airborne point cloud data. For validating the Eye-Amsterdam results, the Dutch building registration dataset BAG (*Basisregistratie Adressen en Gebouwen*) of the 2019 building dataset is used [32]. However, we noticed that several BAG buildings have different shape and size compared to the AHN3 buildings due to different data acquisition time. Thus, the RMSE of the corners is calculated for unchanged building.

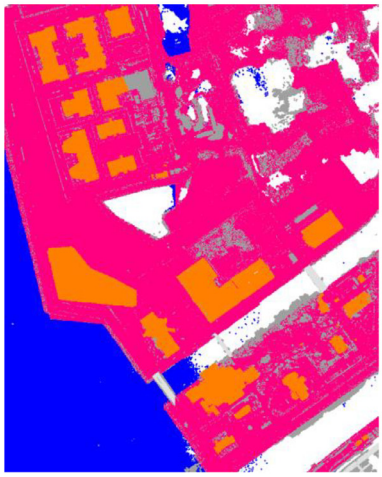
In this study, the required positional accuracy for the outline result is at least 1-meter. For the Makassar dataset, the average RMSE for 36 buildings in the study area is ~ 65 cm, which meets our requirements. There is an exception for one incomplete building (indicated by a red circle in Fig. 11.b.), that has a RMSE of more than one meter. In this case, the building roof is partially covered by dense trees resulting in poor building segmentation. Based on the number of corners from the reference data, the precision, re-



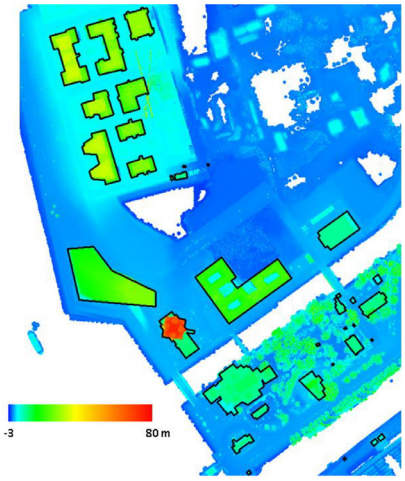
(**a**) Classified point cloud of the Makassar dataset consists of buildings (orange), ground (pink), and vegetation (green). ©BIG.



(**b**) Comparison of the Makassar outline result (black line) to the Digital Surface Model (DSM) image.



(**c**) Classified AHN3 point cloud of the EYE – Amsterdam dataset consists of buildings (orange), ground (pink), water (blue), and unclassified points (gray). ©AHN3.



(**d**) Comparison of the EYE-Amsterdam outline result (black line) to the Digital Surface Model (DSM) image.

Fig. 12. Study areas and building outline results.

call and F1-score of the Makassar test set are 0.99, 0.95, and 0.97 respectively.

Building outline results for the EYE-Amsterdam dataset, as shown in Fig. 12.d, are also good at 0.7 m RMSE on average. For this dataset, there are no classification issues as we use available building points provided by the AHN3. In this case, one of the biggest factors that influences the accuracy of the result is the definition of building roof, in case an overhanging roof exist. This means that the overhanging roof is likely included in a building in AHN-3 data but not in the BAG data, which results in discrepancies. Fig. 13.b shows overhanging roofs on one building in the EYE-Amsterdam area (marked as A in Fig. 13.a). Another issue is found on a building with a curved building outline. The algorithm could not detect points on a curved line, as the normal angle differences δN between edge points is small (less than five degrees). Decreasing the normal angle differences δN threshold would not always mitigate this problem, as it will increase the number of false corners due to noise of the edge points.

4.2. General overview

Compared to our previous study on building outline extraction using ordered points aided Hough transform (OHT) [37], the MAT approach has higher sensitivity to noisy edge points and variations in point density. Small bumps on the edges affect the normal direction of the corresponding point, which later affects the corner detection result. However, the MAT approach has a similar accuracy as our previous work at 0.7 m RMSE for the MAT-based method and 0.57 for the OHT method. Our proposed method is able to handle one of the shrinking ball algorithm limitations that requires the surface normal for each sample point [6].

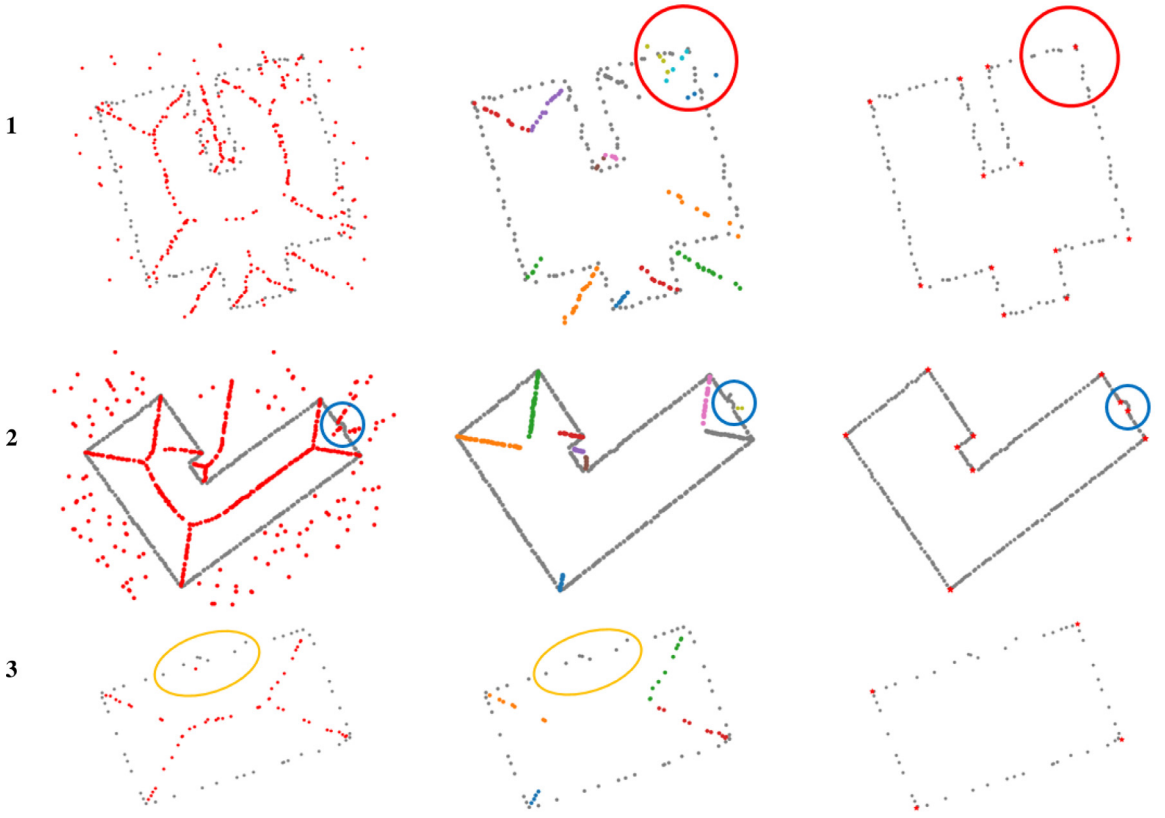
The use of the alpha-shape algorithm makes it possible to orient the normals of each edge point automatically when performing inner and outer MAT computation. Inner and outer MAT points together effectively detect all building corners.

Fig. 14 shows the ability of our method to detect corners of various building shapes with different point density. Fig. 14 row 1 demonstrates how our algorithm works on sparse building edge points that later result in false segments. The red circle in Fig. 14.b row 1 shows three medial segments around a corner. Two of them are considered as false segments. However, the algorithm delivers the correct corner points only (inside the red circle in Fig. 14.c row 1) even though false segments exist. The method also works in case small perturbations exist on the boundary e.g. due to trees (see orange circles in Fig. 14.a row 3). There is one MAT point (the red point inside the yellow ellipse in Fig. 14.a row 3) resulting from a noisy edge part, that is correctly ignored by the medial segmentation procedure. On the other hand, the algorithm may fail

(**a**) Top view building definition of BAG polygon (red) and the AHN3 building points (grey).

(**b**) Building "A" has overhanging roof (red circle). ©GoogleStreetView.

Fig. 13. Comparison of BAG building polygon (red) and the AHN3 building points (grey) in case of overhanging roof.



(**a**) The MAT points (red) of the edge points (gray).

(**b**) Segmented MAT points

(**c**) Resulting corner points (red).

Fig. 14. Outline results for different building shapes.

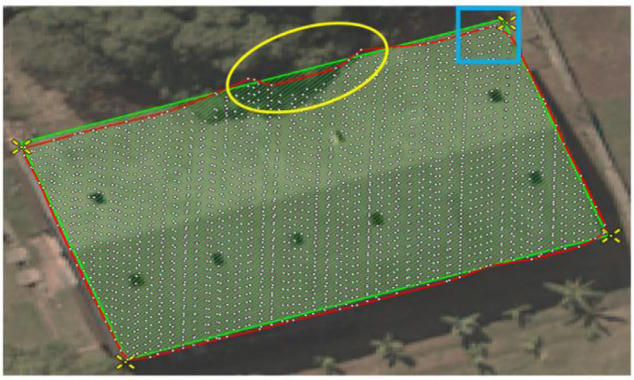
to extract non-linear shapes like rounded edges (see blue circles in Fig. 14. row 2) as the algorithm cannot detect significant normal change for boundary points on the rounded edge. In this case, the algorithm can only detect two candidate corners from two medial segments, positioned at both ends of the rounded edge.

As shown in Fig. 15.a, our method is able to obtain building corners (yellow points) that are close to the reference building polygon (green) even when there is a perturbation in the boundary (inside the yellow ellipse). The proposed method, particularly, improves the alpha-shape outline (red). Fig. 15.b compares between corners from our method, the alpha-shape outline result, and the building outline reference.

In addition, we found out that several MAT properties have not been investigated yet, particularly for the outline extraction, e.g. the curvature of consecutive MAT points indicating an asymmetric shape of two edges of different length, may be used for locating but also characterizing corners accurately.

4.3. Comparison analysis

In this section, we compare results of our proposed method to those of existing methods on building outline extraction. For



a. Corners extracted from our method (yellow) compare to alpha-shape outline (red line) and the building map polygon (green polygon).

b. Zoomed-in of the blue rectangle in Figure 15.a.

Fig. 15. Corner identification comparison between our proposed method and the reference and the alpha-shape outline in Makassar study area.

Table 1
Evaluation of different building outline extraction methods.

Approach	Geometric Accuracy RMSE (m)	Corners Detection Accuracy			Average computational time (*sec*)
		Precision (%)	Recall (%)	F1-score (%)	
Hough-based [20]	0.434	89.48%	95.15%	91.91%	2.10
RanSAC-based [53]	0.548	91.89%	96.14%	93.65%	1.90
MAT-based (the proposed method)	0.414	94.17%	94.55%	93.82%	1.71

this purpose, a small subset of the AHN3 airborne point cloud of Amsterdam is used. The two other building outline extraction methods applied to this test area are: the Ordered Hough Transform (OHT) method proposed by Widyaningrum et al. [37] and a RanSAC-based segmentation and regularization method by Lucas and Van Tilburg [53]. All these methods use an alpha-shape algorithm to select boundary points, which later result in building corners. The RanSAC-based segmentation and regularization method requires primary building orientations as it regularizes all boundary lines with respect to these orientations and its perpendicular orientations. Another regularization approach, OHT [37], applies the so-called extended Hough transform on a list of ordered boundary points that enable to detect arbitrary building directions and extract different boundary segments.

The comparison metrics considered are the building corners geometric accuracy (RMSE), the computation time of building outline extraction after the boundary points are selected, and the corner detection accuracy in terms of recall, precision, and F1-score.

As shown in Table 1., our proposed method has the highest geometric accuracy as well as F-1 score. The RanSAC-based method has higher Recall value as our proposed method, but is likely to have more undetected corners. However, the proposed method has the lowest number of false positive corners. The average computation time for the three methods is considered comparable, although in fact the proposed method is the fastest. However, each method may have different strengths and weaknesses when it is applied on a complex building roof shape. Using the building shown in Fig. 16. as one example, a reference corner inside the brown circle is detected by OHT, but not by the proposed method. The RanSAC-based method results in two corners with an offset close to 1-meter from the reference. The proposed method fails to detect one building corner (inside the brown circle) due to a wide angle between two consecutive building outline segments that is close to 180° In this case, the shrinking circle failed to produce a separation angle α close to 90° needed for corner detection.

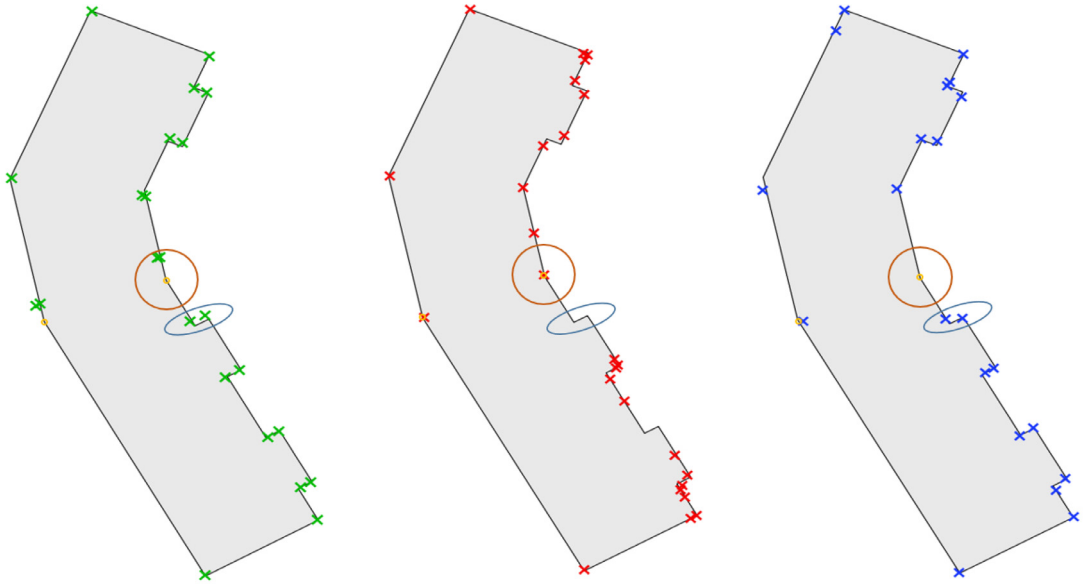
Another difference in the results in Fig. 16. is indicated by the blue circles. The OHT method fails to detect two corners of a short building edge of one meter length, while the RanSAC-based method as well as our proposed method successfully detect both corners. All methods produce false corners but the proposed method has the smallest number of false corners.

4.4. Computational and complexity analysis

We implemented our method in Python2.7 on an Intel Core 2Duo CPU with 2.4 GHz processors. The computation time for our corner detector algorithm varies from 0.28 to 0.99 s per building, depending on the building size, shape and point density.

Recall that the proposed method has four main steps: boundary point selection by an alpha-shape algorithm, MAT extraction by a shrinking circle approach, MAT points segmentation, and estimation of corners by PCA. However, in case building points are not yet available, an additional classification and segmentation step is required. One of the most common method for 3D point cloud segmentation is seeded region growing. This type of algorithms selects a seed point and adds a point from the neighborhood if it meets a certain criterion. As reported by Shih and Cheng [51], and Deschaud and Goulette. [52], the computational complexity of seeded-based region growing for partitioning a point cloud consisting of n 3D points into N segments is $O(n \log n)$. Rabbani et al. [55] improved plane detection using a smoothness constraint based on the normal angle difference between neighboring points and reported a time complexity of $O(n + N\ log\ n)$.

The proposed algorithm works on individual building segments. Suppose a segment contains u number of building points, the computational complexity of our four main steps are as follows:



(**a**) Corners extracted by the RanSAC-based method (green crosses).

(**b**) Corners extracted by the OHT method (red crosses).

(**c**) Corners extracted by the proposed method (blue crosses).

Fig. 16. Building corners resulting from three different methods compared to building reference (gray polygon). Results notably vary within the brown circle and blue ellipse.

1. The alpha-shape algorithm for selecting boundary points using Delaunay triangulation, as reported by [58], has a time complexity of $O(u \log u)$.
2. Given v boundary points extracted by, e.g. the alpha-shape algorithm, the shrinking circle algorithm used a KD-Tree to search its nearest point with time complexity of $O(v \log v)$. The shrinking circle algorithm has a computational complexity of $O(v^2)$ because the algorithm may have to visit all medial axis points $\boldsymbol{c_p}$ for every boundary point [6].
3. Medial axis points segmentation has a time complexity of $O(v^2)$, but performance in practice is close to $O(v)$ [6]. Each MAT point m is checked for its median value based on its corresponding boundary points (p_{idx}, q_{idx}) which takes $O(v)$ time. The selection of candidate corner points based on the angular differences between two neighboring boundary points (p, q) takes $O(v)$ as well. The selection of MAT points m belonging to the same segment based on the similarity of median value of two boundary points (p, q) and corner point indices also has a time complexity of $O(v^2)$, which leads to an overall time complexity of $O(v^2)$.
4. The overall time complexity of PCA is $O(d^3 + d^2 m)$ where m is the number of sample points and d is the number of features [59]. Here, $d = 2$. For estimating the corner position corresponding to zero radius on a line fitted by the PCA, the time complexity is therefore linear in the number of sample points.

If all points of a building segment were boundary points, the computational complexity would be $O(v^2)$ time. This computational complexity for the total procedure of medial axis segmentation is not a problem because the algorithm works just on the boundary points of a single building. Point cloud segmentation is in practice the heaviest computation as it involves the complete 3D point cloud data. For example, in our Makassar test area, the building segmentation step considered all 464.191 points, while the medial axis segmentation considers 43 individual buildings. The number of points per-building in the Makassar test area varies from 96 to 3185 points, that later results in a number of building boundary points varying from 23 to 156 points. Note that the processing of individual buildings can also be parallelized easily.

5. Conclusions and recommendations

In this study, we have presented a procedure for automatically extracting building outlines from airborne point clouds based on the MAT descriptors generated by the 2D shrinking circle method. Our approach takes advantage of MAT invertibility with its medial axis and the corresponding radius function that allows reconstructing the exact object shape. Building classification is conducted first. A set of building boundary points is then extracted using an alpha-shape algorithm. After applying a shrinking circle algorithm to the input boundary points, MAT points are obtained. To identify corners, we introduce a corner-aware segmentation to group MAT points to their corresponding medial branch. The segmentation combines both global thresholds and several MAT-derived features. Next, the algorithm fits a line to all MAT points of a segment. Based on the corresponding radii of the MAT points, the corner point location is estimated by extrapolating the position where the radius is zero on the fitted line.

The positional accuracy results of the estimated corner points indicate that our method provides a completely new and promising tool for reconstructing the geometric shape of building roofs from scattered airborne point clouds by using the MAT approach. The proposed method performance is highlighted on a number of complex building shapes in and around the EYE building in Amsterdam, The Netherlands. The ability of the proposed method to obtain accurate corners and complete shapes indicates the robustness of our method to small perturbations on the building edge. In case of sparse point intervals, densification of edge points may help to increase the accuracy of the MAT result. Meanwhile, our method has limitations to obtain outlines of an object with a curved or circular shape. In comparison to state-of-the-art methods on building roof outline extraction, our proposed method shows a promising result in acquiring accurate building corners geometrically. Compare to RanSAC and Hough transform based methods,

our method is a primitives-free approach that does not require orientation initialization.

Though current skeletonization methods show a progressive development, deployment for wider applications is still challenging. Different applications may require different skeletonization methods and/or MAT descriptors. For future work, we will consider extending the MAT technique for reconstructing other digital map objects, such as road networks, ridges, or streamlines. The application of the proposed workflow for extracting curved lines and for a larger area is also interesting to be investigated further.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

The authors acknowledge Indonesian Geospatial Agency (www.big.go.id) for providing the Makassar datasets in this research. The authors gratefully acknowledge the Indonesia Endowment Fund for Education (LPDP), Ministry of Finance of Republic of Indonesia for a scholarship support to the first author.

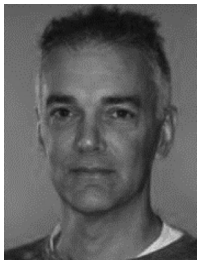
References

[1] T. Stavros, S. Dickinson, AMAT: mmedial axis transform for natural images, Proc. IEEE Int. Conf. Comput. Vis. (2017), doi:10.1109/ICCV.2017.295.
[2] R. Wang, J. Peethambaran, D. Chen, LiDAR point clouds to 3-D urban models: aa review, IEEE J. Sel. Top. Appl. Earth Observ. Remote Sens. 11 (2) (February 2018), doi:10.1109/JSTARS.2017.2781132.
[3] R. Peters, Geographical Point Clouds Modelling with the 3D Medial Axis Transform (2018) the.
[4] A. Tagliasacchi, T. Delame, M. Spagnuolo, N. Amenta, A. Telea, 3D skeletons: aa state-of-the-art report, Comput. Graphic. Forum 35 (2) (May. 2016) 573–597 pp, doi:10.1111/cgf.12865.
[5] H. Blum, A transformation for extracting new descriptors of shape, Model. Percept. Speech Visual Form 19 (5) (1967) 362–380.
[6] J. Ma, S.W. Bae, A. Choi, 3D medial axis point approximation using nearest neighbors and the normal field, Visual Comput. 28 (2012) 7–19, doi:10.1007/s00371-011-0594-7.
[7] D. Reniers, J.J. Van Wijk, A. Telea, Computing multiscale skeletons of genus 0 objects using a global importance measure, IEEE Trans. Visual. Comput. Graph. 14 (2) (2008) 355–368, doi:10.1109/TVCG.2008.23.
[8] J. Giesen, B. Miklos, M. Pauly, C. Wormser, The scale axis transform, in: Proceedings of the 25th Annual Symposium on Computational Geometry, Aarhus, Denmark, June 2009, pp. 106–115, doi:10.1145/1542362.1542388.
[9] X. Bai, L. Latecki, W. Liu, Skeleton pruning by contour partitioning with discrete curve evolution, IEEE Trans. Pattern Anal. Mach. Intell. 29 (3) (2007) 449–462, doi:10.1109/TPAMI.2007.59.
[10] O.K.C. Au, C. Tai, H. Chu, D. Cohen-Or, T. Lee, Skeleton extraction by mesh contraction, in: Proc. ACM SIGGRAPH, 2008, pp. 441–449, doi:10.1145/1360612.1360643.
[11] F. Leymarie, B. Kimia, The medial scaffold of 3d unorganized point clouds, IEEE Trans. Pattern Anal. Mach. Intell. 29 (2) (2007) 313–330, doi:10.1109/TPAMI.2007.44.
[12] P.K. Saha, G. Borgefors, G. Sanniti di Baja, A survey on skeletonization algorithms and their applications, Pattern Recognit. Lett. 76 (2016) 3–12, doi:10.1016/j.patrec.2015.04.006.
[13] P.K. Saha, G. Borgefors, G. Sanniti di Baja, Skeletonization and its applications – A review, in: Skeletonization: Theory, Methods and Applications, 2017, pp. 3–42, doi:10.1016/b978-0-08-101291-8.00002-x.
[14] J. Chaussard, M. Couprie, H. Talbot, in: A Discrete λ-Medial Axis, Springer, 2009, pp. 232–243, doi:10.1007/978-3-642-04397-0_36. *Proc. DGCI*.
[15] J. Kustra, A. Jalba, A. Telea, Computing refined skeletal features from medial point clouds, Pattern Recognit. Lett. (2015), doi:10.1016/j.patrec.2015.05.007.
[16] N. Ahuja, J. Chuang, Shape representation using a generalized potential field model, IEEE Trans. Pattern Anal. Mach. Intell. 19 (2) (1997) 169–176, doi:10.1109/34.574801.
[17] T. Pavlidis, A thinning algorithm for discrete binary images, Comput. Graph. Image Process. 13 (1980) 142–157 pp.
[18] D. Lee, Medial axis transformation of a planar shape, IEEE Trans. Pattern Anal. Mach. Intell. 4 (1982) 363–369, doi:10.1109/TPAMI.1982.4767267.
[19] T. Culver, J. Keyser, D. Manocha, Exact computation of the medial axis of a polyhedron, Computer Aided Geometric Design 21 (1) (2004) 65-98, doi:10.1016/j.cagd.2003.07.008.
[20] E. Widyaningrum, B. Gorte, R.C. Lindenbergh, Extraction of building roof edges from LiDAR data to optimize the digital surface model for true orthophoto generation, Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci. 42 (2) (2018), doi:10.5194/isprs-archives-XLII-2-1199-2018.
[21] T. Pavlidis, A review of algorithms for shape analysis, Comput. Graph. Image Process. 7 (2) (1978) 243–258 pp, doi:10.1016/0146-664X(78)90115-6.
[22] C.M. Holt, A. Stewart, M. Clint, R.H. Perrott, An improved parallel thinning algorithm, Commun. ACM 29 (3) (1987) 239–242, doi:10.1145/12527.12531.
[23] M. Couprie, G. Bertrand, Asymmetric parallel 3d thinning scheme and algorithms based on isthmuses, Pattern Recognit. Lett. 76 (2016) 22–31, doi:10.1016/j.patrec.2015.03.014.
[24] N. Amenta, C.S. Choi, and R.K. Kolluri. The power crust, In *Proceedings of the sixth ACM Symposium on Solid modeling and Applications*, pp. 249-266.2001, https://dl.acm.org/doi/10.1145/376957.376986
[25] H. Huang, S. Wu, D. Cohen-Or, M. Gong, H. Zhang, G. Li, B. Chen, L1-medial skeleton of point cloud, ACM Trans. Graph. 32 (4) (2013) pp65–1, doi:10.1145/2461912.2461913.
[26] T. Broersen, R. Peters, H. Ledoux, Automatic identification of watercourses in flat and engineered landscapes by computing the skeleton of a LiDAR point cloud, Comput. Geosci. 106 (2017) 171–180, doi:10.1016/j.cageo.2017.06.003.
[27] LAStools, Efficient LiDAR processing software (version 141017, academic), obtained from http://rapidlasso.com/LAStools, accessed on October 2018.
[28] H. Edelsbrunner, D. Kirkpatrick, R. Seidel, On the shape of a set of points in the plane, IEEE Trans. Inf. Theory 4 (1983) 551–558 IT-29, doi:10.1109/TIT.1983.1056714.
[29] M. Yirci, M. Bredif, J. Perret, N. Paparoditis, 2D Arrangement-based hierarchical spatial partitioning: aan application to pedestrian network generation, in: Proceedings of the Sixth ACM SIGSPATIAL International Workshop on Computational Transportation Science, 2013, pp. 31–36, doi:10.1145/2533828.2533843.
[30] J.H. Haunert, M. Sester, Area collapse and road centerlines based on straight skeletons, GeoInformatica 12 (2) (2008) 169–191, doi:10.1007/s10707-007-0028-x.
[31] AHN3 point clouds data, obtained from https://www.pdok.nl/nl/ahn3-downloads, accessed on October 2018.
[32] BAG data -. Gebouwen (INSPIRE geharmoniseerd), obtained from http://geodata.nationaalgeoregister.nl/bag/wfs?service=WFS&request=GetCapabilities, accessed on May 2019.
[33] M. McAllister, J. Snoeyink, "Medial axis generalization of river networks", Cartogr. Geograph. Inf. Sci. 27 (2) (2000) 129–138, doi:10.1559/152304000783547966.
[34] N. Regnauld, W.A. Mackaness, "Creating a hydrographic network from its cartographic representation: aa case study using ordnance survey Master Map data", Int. J. Geograph. Inf. Sci. 20 (6) (2006) 611–631 https://doi.org/10.1080/13658810600607402.
[35] J. Chen, L. Zhou, J. Zhang, L. Dou, "The comparison and application of corner detection algorithms", J. Multimedia 4 (6) (2009).
[36] P. Ghosh, A. Pandey, U.C. Pati, Comparison of different feature detection techniques for image matching, ACCENTS Trans. Image Process. Comput. Vis. 1 (1) (2015), doi:10.4304/jmm.4.6.435-441.
[37] E. Widyaningrum, B. Gorte, R. Lindenbergh, Automatic building outline extraction from ALS point clouds by ordered points aided Hough transform, Remote Sens. 11 (14) (2019) 1727, doi:10.3390/rs11141727.
[38] E. Widyaningrum, R. Lindenbergh, Skeleton-based road network extraction using an orthophoto colored point cloud, in: *Proceedings of the* 40th *ACRS*, 2019.
[39] T.Y. Zhang, C.Y. Suen, A fast parallel algorithm for thinning digital patterns, Commun. ACM 27 (3) (1984), doi:10.1145/357994.358023.
[40] C.W. Niblack, P.B. Gibbons, D.W. Capson, Generating skeletons and centerlines from the distance transform, CVGIP 54 (5) (1992) 420–437, doi:10.1016/1049-9652(92)90026-T.
[41] Y. Ge, J.M. Fitzpatrick, On the generation of skeletons from discrete Euclidean distance maps, IEEE Trans. Pattern Anal. Mach. Intell. 18 (11) (1996) 1055–1066, doi:10.1109/34.544075.
[42] W.P. Choi, K.M. Lam, W.C. Siu, Extraction of the Euclidean skeleton based on a connectivity criterion, Pattern Recognit. 36 (2003) 721–729, doi:10.1016/S0031-3203(02)00098-5.
[43] J. Chuang, C. Tsai, M.C. Ko, Skeletonization of three-dimensional object using generalized potential field, IEEE Trans. Pattern Anal. Mach. Intell. 22 (11) (2000) 1241, doi:10.1109/34.888709.
[44] T. Grigorishin, Y.H. Yang, Skeletonization: an electrostatic field-based approach, Pattern Anal. Appl. 1 (1998) 163–177.
[45] H. Edelsbrunner, D.G Herbert, Kirkpatrick, R. Seiderl, On the shape of a set of points in the plane, IEEE Trans. Inf. Theory 29 (4) (1983) 551–559, doi:10.1109/TIT.1983.1056714.
[46] C. Zheng, J. Wang, B. Lehrbass, An evaluation system for building footprint extraction from remotely sensed data, IEEE J. Sel. Top. Appl. Earth Observ. Remote Sens. 6 (3) (2013) 1640–1652, doi:10.1109/JSTARS.2013.2256882.
[47] J.M. Reddy, G.M. Turkiyyah, Computation of 3D skeletons using a delaunay triangulation technique, Comput. Aided Des. 27 (9) (1995) 677–694.
[48] F.H. She, R.H. Chen, W.M. Gao, P.H. Hodgson, L.X. Kong, H.Y. Hong, Improved 3D thinning algorithms for skeleton extraction, IEEE 2009 Digit. Image Comput. (2009), doi:10.1109/DICTA.2009.13.
[49] J.R. Taylor, S.T. Lovell, Mapping public and private spaces of urban agriculture in Chicago through the analysis of high-resolution aerial images in Google Earth, Landscape Urban Plann. 108 (1) (2012) 57–70, doi:10.1016/j.landurbplan.2012.08.001.

[50] M. Ester, H.P. Kriegel, J. Sander, X. Xu, A density-based algorithm for discovering clusters in large spatial databases with noise, KDD-96 Proc. (1996).
[51] F.Y. Shih, S. Cheng, Automatic seeded region growing for color image segmentation, Image Vis. Comput. 23 (10) (2005) 877–886, doi:10.1016/j.imavis.2005.05.015.
[52] J.E. Deschaud and F. Goulette, A fast and accurate plane detection algorithm for large noisy point clouds using filtered normals and voxel growing, 3DPVT, May 2010, Paris, France. 2010.
[53] C. Lucas, and T. van Tilburg, Building Boundary. 2019. Retrieved Jan 16, 2020, from https://github.com/Geodan/building-boundary
[54] A. Manno-Kovacs, T. Sziranyi, Orientation-selective building detection in aerial images, ISPRS J. Photogramm. Remote Sens. 108 (2015) 94–112, doi:10.1016/j.isprsjprs.2015.06.007.
[55] Z. Zhao, Y. Duan, Y. Zhang, R. Cao, Extracting buildings from and regularizing boundaries in airborne LiDAR data using connected operators, Int. J. Remote Sens. 37 (2016) 889–912, doi:10.1080/014311611.2015.1137647.
[56] M. Awrangjeb, Using point cloud data to identify, trace, and regularize the outlines of buildings, Int. J. Remote Sens. 37 (2016) 551–579, doi:10.1080/01431161.2015.1131868.
[57] Y. Li, H. Wu, R. An, H. Xu, Q. He, J. Xu, An improved building boundary extraction algorithm based on fusion of optical imagery and LIDAR data, Optik (Stuttg) 124 (2013) 5357–5362.
[58] C. Varytimidis, K. Rapantzikos, Y. Avrithis, S. Kollias, α-shapes for local feature detection, Pattern Recognit. 50 (2016) 56–73, doi:10.1016/j.patcog.2015.08.016.
[59] J. Makhoul, F. Kubala, R. Schwartz, R. Weischedel, Performance measures for information extraction, in: DARPA Broadcast News Workshop, 1999, pp. 249–252.
[60] E.K. Forkuo, B. King, Automatic fusion of photogrammetric imagery and laser scanner point clouds, Int. Arch. Photogramm. Remote Sens. 35 (4) (2004) 921–926.
[61] S. Devi, Veena, Measurement of relief displacement from vertical photograph, Int. J. Sci. Eng. Technol.Res. 3 (10) (2014).

Elyta Widyaningrum is currently a PhD student at the Department of Geoscience and Remote Sensing, Delft University of Technology, the Netherlands. She also with the Geospatial Information Agency, Indonesia. Her research interests are in the field of machine learning, object extraction, and segmentation/classification from point cloud data.

Ravi Y. Peters is currently a post-doctoral researcher at the Department of Urbanism, Faculty of Architecture and the Built Environment, Delft University of Technology, the Netherlands. His-main research interests are point cloud analysis and 3D reconstruction.



Roderik C. Lindenbergh is Associate Professor at the Department of Geoscience and Remote Sensing, Delft University of Technology, the Netherlands. He studied Mathematics and obtained a PhD on geometry and topology from Utrecht University. His-main research interests are laser scanning and geometric information extraction from spatial point clouds.