

## Extraction of building roof edges from LiDAR data to optimize the digital surface model for true orthophoto generation

Widyaningrum, E.; Lindenbergh, R. C.; Gorte, B. G.H.; Zhou, K.

**DOI**

[10.5194/isprs-archives-XLII-2-1199-2018](https://doi.org/10.5194/isprs-archives-XLII-2-1199-2018)

**Publication date**

2018

**Document Version**

Final published version

**Published in**

International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives

**Citation (APA)**

Widyaningrum, E., Lindenbergh, R. C., Gorte, B. G. H., & Zhou, K. (2018). Extraction of building roof edges from LiDAR data to optimize the digital surface model for true orthophoto generation. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives*, 42(2), 1199-1205. <https://doi.org/10.5194/isprs-archives-XLII-2-1199-2018>

**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.

# EXTRACTION OF BUILDING ROOF EDGES FROM LIDAR DATA TO OPTIMIZE THE DIGITAL SURFACE MODEL FOR TRUE ORTHOPHOTO GENERATION

E. Widyaningrum<sup>1,2\*</sup>, R.C. Lindenbergh<sup>1</sup>, B.G.H. Gorte<sup>3</sup>, K. Zhou<sup>1</sup>

<sup>1</sup> Dept. of Geoscience and Remote Sensing, Delft University of Technology, The Netherlands - (e.widyaningrum, r.c.lindenbergh)@tudelft.nl

<sup>2</sup> Centre for Topographic Base Mapping and Toponyms, Geospatial Information Agency, Indonesia – elyta.widyaningrum@big.go.id

<sup>3</sup> University of New South Wales, Australia – b.gorte@unsw.edu.au

Commission II, WG II/4

**KEY WORDS:** LiDAR, Aerial photos, Edge Detection, DSM, True Orthophoto

## ABSTRACT:

Various kinds of urban applications require true orthophotos. True orthophoto generation requires a DSM (Digital Surface Model) to project the photo orthogonally and minimize geometric distortion due to topographic variance. DSMs are often generated from airborne laser scan data. In urban scenes, DSM data may fail to deliver sharp and straight building roof edges. This will affect the quality of the resulting orthophotos. Therefore, it is necessary to incorporate good quality building outlines as breaklines during DSM interpolation. This study proposes a data-driven approach to construct building roof outlines from LiDAR point clouds by a workflow consisting of the following steps: given roof segments, roof boundary points are extracted using a concave hull algorithm. Straight edges may be difficult to find in complex roof configurations. Therefore, two ingredients are combined. First, RanSAC corner point preselection, and second, DBSCAN-based clustering of edge points. The method is demonstrated on an area of  $\pm 1.2$  km<sup>2</sup> containing 42 buildings of different characteristics. A quality assessment shows that the proposed method is able to deliver 92% of building lines with acceptable geometric accuracy in comparison to a building line in the base map.

## 1. INTRODUCTION

### 1.1 Background

Many urban applications require true orthophotos as a basic geospatial data source to construct real-world visualizations with accurate positions. Buildings are one important objects to be presented on the map. High-elevated buildings may suffer from positional distortion due to relief displacement caused by the aerial photo acquisition geometry. True Orthophoto generation is one of the most reliable methods to remove distortion by projecting back image pixels to the correct location using a DSM. However, because of DSM gridding, some buildings may lose their sharp edges. This will introduce building boundary errors in the true orthophotos (the building edges looks crooked or bent in the image). A possible solution is to add building outlines as breaklines to the DSM to preserves sharp building edges. The suitability of true orthophotos to define object edges relies on the quality of the DSM. The sharper the building edges are represented in the DSM, the better the building edge representation will be in the resulting true orthophotos.

In the absence of base maps or updated maps, it is required to extract building edges. LiDAR data is known for its high 3D positional accuracy and high point density for mapping. In the case of boundary detection, LiDAR may have a limitation in providing sharp edges, as LiDAR points, in general, do not always hit the exact building edges.

This paper proposes and implements a semi-automated workflow to extract building roof edges from LiDAR data by combining an adapted version of RanSAC (RANdom SAMple Consensus) line fitting with DBSCAN (Density-Based Spatial Clustering of Applications with Noise) clustering of point-wise line parameters.

### 1.2 Related Work

Some true orthophoto refinement efforts that use breaklines for DSM improvement have already been conducted (Wang et al., 2018, Hu et al., 2016, Wu et al., 2011, Kato et al., 2010). Wu et al. (2011) improved building boundaries by fusing LiDAR and aerial photos to get 3D lines. The 3D building boundaries were extracted from segmented LiDAR points using a convex hull algorithm followed by a back-projection of 3D to 2D image space. However, many buildings boundaries are not as straight as expected.

Hu et al. (2016) worked on orthophotos with occluded areas detected by elevation-buffer visibility from DSM and building roof model. Then, occluded areas are filled with data from overlapped orthophotos. This method still has limitations since the results may be not good along building boundaries in case of inaccurate building roof models. Wang et al. (2018) proposed to match 2D line segments to reconstruct 3D lines. The 3D end points of each line are then incorporated with dense image matching points clouds in a TIN (Triangular Irregular Network) processing.

Automatic building roof outline extraction is still a challenging problem, especially when straight and accurate edges are required. Several types of research on automatic building edge extraction from either LiDAR data and images have been conducted (Awrangjeb, 2016, Zhao et al., 2016, Siddiqui, et al., 2016, Yang et al., 2013, Wu et al., 2011, Awrangjeb, 2010). Wu et al. (2011) fused rough building lines from a LiDAR DSM and 2D lines from oriented aerial images to extract straight 3D building lines. These straight building lines are then used to produce true orthophotos. However, some resulting building boundaries were still not straight. Zhao et al. (2016) and Awrangjeb (2016) determined building outlines from point cloud

data by boundary tracing and regularization. Somehow, errors occur due to missing low buildings and vegetation points near to the buildings Zhao et al. (2016), or failures to estimate a dominant/principal direction Awrangjeb (2016). Siddiqui et al. (2016) proposed a gradient-based technique to extract building lines from LiDAR and images. Roof edges correspond to big height changes that are revealed by large gradients in a height intensity image. Straight lines are then extracted from the image using Canny and Gaussian functions. The buildings outlines were confirmed after color matching, shadow detection, and noise filtering. The proposed method is only designed for buildings with a flat or sloped roof. However, the paper did not address the possible relief displacement on the building roof may occur in the image space. A common problem of model-driven approaches is that some important lines that are not dominant may not be found. Moreover, the quality of geometric accuracy of these methods is not reported and results often seem incomplete.

## 2. STUDY AREA AND DATA DESCRIPTION

The chosen suburban study area is located on Maros, Sulawesi island of Indonesia. The total size of the study area is  $\pm 1.2$  km<sup>2</sup>. There are two main components of data used in this study:

### a. Main input data

LiDAR data was acquired by a Leica ALS70 instrument. The average point density is 10 - 12 points/m. A Leica RCD30 instrument captured the aerial photos with 9cm spatial resolution. The data use the WGS84 reference system and is projected in UTM Zone 50S. Aerial photos and LiDAR data are acquired at the same time from the same platform in 2012. The specification of these main input data is designed and intended for base mapping at a scale of 1:10.000.

### b. Reference data

This study uses 1:10.000 Topographic Base Maps as a reference to validate the accuracy of the resulted lines and the true orthophotos. The topographic base map has a geometric accuracy better than 5 meters. The reference data consists of 3D building rooflines that are manually delineated from stereo-photos acquired at the same time as the LiDAR data.

## 3. METHODOLOGY

The research workflow, as illustrated in Figure 1., is designed to extract building roof outlines to improve the quality of DSM for true ortho-rectification without any presence of base maps.

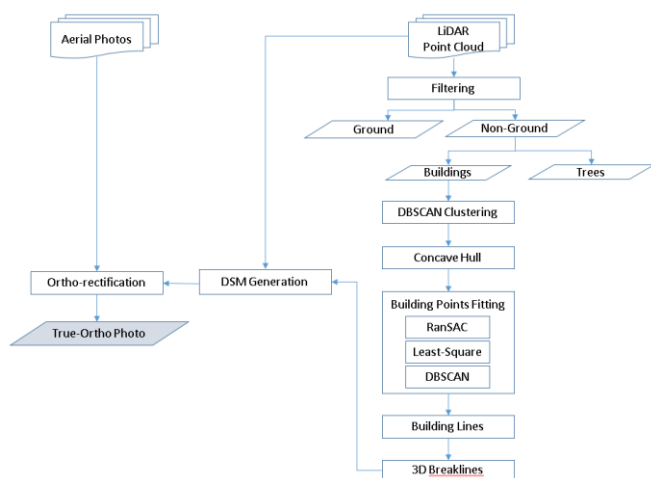


Figure 1. Methodological workflow

This study aims to extract building roof outlines that meet specifications and assumptions as specified below:

- The building is represented by the size and shape of a 2D representation of the roof. That is, the size and shape of the buildings are similar to the roof as seen from the top.
- Small details or interior parts inside the building roof will not be considered. For example, in extracting the outlines, a building with a gable roof will be treated as a flat roof.
- The expected result should meet the requirements of 1:5.000 map scale. A building of an area of 2.5 m x 2.5 m or larger should be present in the map.
- The minimum positional accuracy of building edges vectors is 2.5 meters.

### 3.1 LiDAR Point Cloud Filtering

LiDAR point cloud filtering is the first important step to guarantee the quality of the extracted buildings edges. This study uses a set of building points extracted by a rule-based approach in the TerraScan software as a starting point. In rule-based filtering, the first step is to separate non-ground from ground points. The roof points are then selected from the non-ground points based on their planarity and connectivity. The result building filtering of the study area is shown in Figure 2.

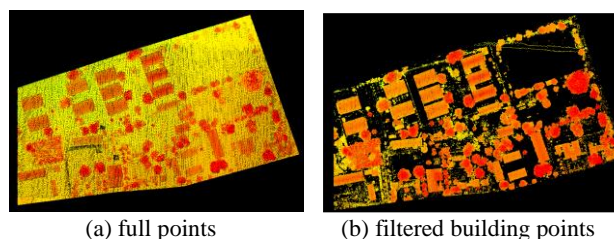


Figure 2. LiDAR point clouds of the study area.

### 3.2 Building Points Clustering and Edge Point Selection

In order to find roof edge points, roofs are identified by segmentation. This study uses DBSCAN, introduced by Ester et al. (1996), to cluster points into single buildings. Ghosh and Lohani (2013) performed a DBSCAN on a small subset of LiDAR data (100 m by 100m) with a high success rate w.r.t. computation time and accuracy.

After clustering some noise or outliers (including tree points) may remain that will finally affect the roof edge extraction. DBSCAN is chosen because of its capability to detect outliers in the data.

To find a cluster, DBSCAN starts with an arbitrary seed point  $p$  and retrieves all neighbouring points (density-reachable) from  $p$  that are located within a given radius distance ( $eps$ ) and contains a given minimum number of points ( $minPts$ ). Outliers are defined once  $minPts$  cannot be achieved within the given  $eps$ . DBSCAN uses region-growing to grow the cluster. The cluster will grow as long as nearby points within the  $eps$  distance from seed  $p$  fulfill the  $minPts$  threshold. In case  $minPts$  within distance  $eps$  is not fulfilled, a point or group of points is considered as outliers. During the cluster growing, outliers may change into a member of the cluster once they are in the  $eps$  distance from the respective seed. To grow the next cluster, the next seed is chosen that does not belong to any cluster. The clustering stops once all points are assigned.

In this study, we implement DBSCAN clustering from 3D and 2D point data. By applying a similar parameter threshold, we found that clustering 3D point data discovers more buildings than 2D points as discussed in Figure 3. Therefore, we used 3D points as input for the edge line extraction.

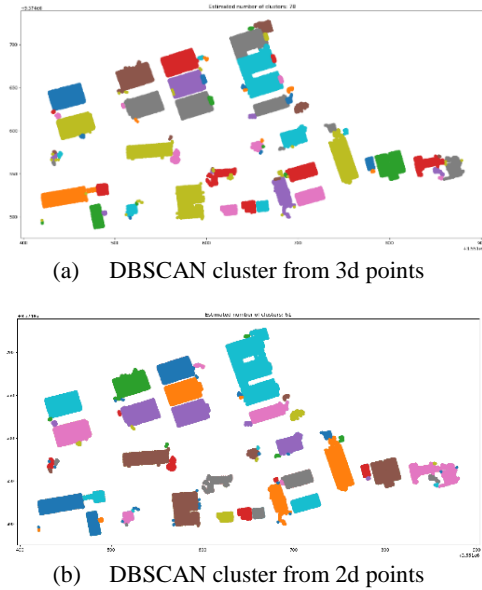


Figure 3. The DBSCAN result using  $eps=1.2m$  and  $minPts=3$

From one building segment, ideally corresponding to a roof, a concave-hull algorithm is used to find the outermost points of that segment. This study uses a generalization of the convex hull, named alpha-concave hull modified by Asaeedi et al. (2003). It uses alpha to define the smoothness threshold. In order to preserve the building shape, the alpha threshold parameter of the concave hull may need to be adapted especially for noisy data.

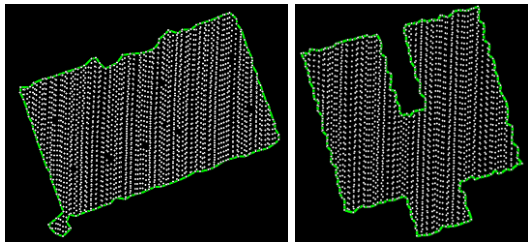


Figure 4. The concave hull line of the buildings

Even though the building outlines obtained by the concave hull method somehow fit the roof lines, they still do not meet the requirements of producing a straight line.

### 3.3 Building lines extraction

The workflow design assumes that most buildings roofs consist of straight lines instead of curved lines. Ransac has the ability to fit a mathematical primitive to set of data containing noise. This algorithm was introduced by Fischler and Bolles in 1981.

In this study, Ransac is employed to find the initial line models through all the boundary points. Ransac typically finds the edge sampled by most points first. Since the Ransac line model is influenced by the selection of random points, we guide Ransac better to find suitable lines. Instead of using two random points, we use one initial point and one random point. To find the given initial points, we compute the mean position (centroid) of all points and calculate the Euclidean distance of each point to the centroid. The farthest point from the centroid is typically one of the corner points of the cluster. This is then used as one of the two points for line fitting in each Ransac run.

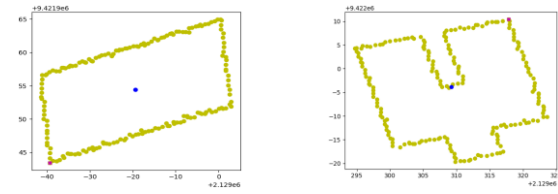


Figure 5. The centroid (blue) and corner points (red)

The Ransac parameters used in this study are minimum sample size ( $minS$ ), distance to the model threshold ( $d$ ), inlier ratio ( $r$ ), and number of iterations ( $k$ ). To initialize a line, two points are chosen from the given samples  $s$  to define a line model. Once the line model is created, the distance of each point to the respective line model is calculated. The final fit is obtained by fitting these inliers to the line model in the sense of least squares. For each edge, we use 100 iterations to find the potential line. The line model that exceeds the predefined  $r$  and has most inliers will be selected. The Ransac parameters are tuned based on the expected requirements and input data specifications. In a normal case, for a point cloud with 50 cm spacing, we use an inliers ratio of 0.3 and a 0.6 meter model threshold distance.

#### Procedure 1: Line Fitting using Adaptive Ransac

Given inliers\_ratio ( $r$ ), no of iterations ( $k$ ), ransac\_threshold ( $d$ ), and dataset ( $D$ ).

```

1 AdaptiveRansac  $L_p(D, k, d, r)$ 
2   for all  $p$  in  $D$ :
3     for  $k^{th}$  iteration do:
4       Calculate centroid ( $c$ )
5       Calculate the farthest point ( $f$ ) to the ( $c$ ).
6       Combine random point ( $R$ ) and  $f$ 
7       Create line models ( $L_1$ ) of  $R$ 
8     return  $m, c$ 
9   for any  $p$  in  $D$ :
10    Find intersection point from  $L_1$ 
11    Compute distance ( $dist$ ) of  $p$  to  $L_1$ 
12    If  $dist(p)$  to  $L_1 < d$ :
13      list_p = inliers of  $L_1$ 
14      If number of inliers  $num > minS * r$ :
15         $L_p = L_1$ 
16  return  $m_1(L_1), c_1(L_1), list\_p1(L_1)$ 
17      # line model is found
18  Given optimum distance threshold ( $d_{opt}$ )
19  for any  $p$  in  $D$ :
20    Compute  $d(p)$  to  $L_1$ 
21    If  $d(p)$  to  $L_1 \leq d_{opt}$ 
22       $p = list\_p2(L_1)$ 
23     $list\_p3(L_1) = list\_p1(L_1) + list\_p2(L_1)$ 
24     $m_2(L_1), c_2(L_1) = leastsquare\_fit(list\_p3(L_1))$ 
25  return  $m_2(L_1), c_2(L_1), list\_p3(L_1)$ 
     $D_2 = D - list\_p3$ 

```

Do step 2 to 25 for the  $D_2$  dataset (to find edge-2)

In case of noisy edge points, Ransac may still fail to find enough potential points as inliers. Therefore, as an additional step the distance threshold is relaxed if needed. The resulting new inlier points are then added to the existing inliers, as already found by Ransac. On the final set of inliers, we apply least-square line fitting to get a robust first building edge. For the second edge, we remove all inliers points belonging to the first line from the dataset. Then, for the remaining points, we apply the same adaptive Ransac procedure. This procedure is repeated until all building edge lines are found (Figure 5.). Given the building lines, corner points are computed based on the line intersections.

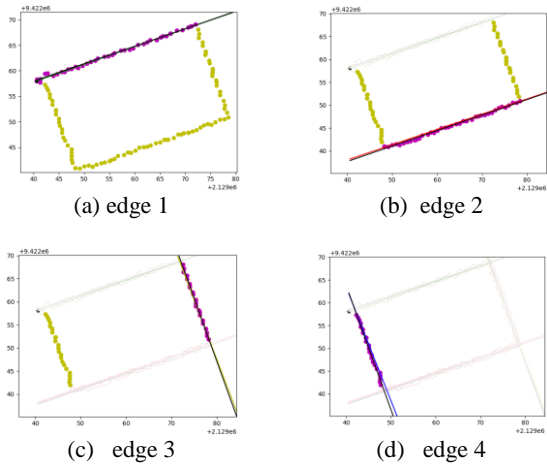


Figure 5. Adaptive RansAC procedure for a simple roof

Any edge extraction method should accommodate variations of building size and shape. Based on our experiments, the proposed adaptive RansAC approach perfectly works for simple buildings (rectangular shape) with less noise but may fail to find lines for more complex buildings (multi-square shape) that have more noisy edge points also.

To accommodate complex roofs, we further adapted RansAC by combining it with DBSCAN clustering. DBSCAN is used to cluster 2D edge points to give direction to RansAC to find inliers and create suitable lines.

**Procedure 2:** Line Fitting combined Adaptive RansAC and DBSCAN

Given inliers\_ratio ( $r$ ), no of iterations ( $k$ ), ransac\_threshold ( $d$ ), dataset ( $D$ ), minimum cluster radius ( $eps$ ), and minimum number of points ( $minSamples$ ).

```

1  Ransac-DB  $L_p(D, k, d, r, eps, minSamples)$ 
2  for all  $p$  in dataset  $D$ :
3      do AdaptiveRansac  $L_p(D, k, d, r)$ 
4      return  $m_2(L_1), c_2(L_1), list\_p3(L_1)$ 
5       $D_2 = D - list\_p3(L_1)$ 
6      for  $D_2$  dataset:
7           $db = dbscan(eps, minSamples)$ 
8          for  $i$  in  $db$ :
9              if  $count(db(i)) = max$ :
10                  $(db(i)) = biggest\_cl$ 
11             return  $biggest\_class$ 
12             for all  $p$  in dataset  $biggest\_cl$ :
13                 do AdaptiveRansac  $L_p(biggest\_cl, k, d, r)$ 
14                 return  $m_2(L_2), c_2(L_2), list\_p3(L_2)$ 
15              $D_3 = D - list\_p3(L_2)$ 
    
```

Do step 1 to14 for the  $D_3$  dataset (to find edge-3)

The second procedure is started by line fitting using adaptive RansAC as described in procedure 1. In most cases, RansAC will be able to find the longest edge since it has most points. Once the corresponding line is found, we remove its supporting inliers from the dataset. Then, we apply DBSCAN to cluster the remaining points. The biggest cluster is selected first. The RansAC procedure is then applied to this cluster to find the second line. Again, the iteration stops once all lines are found.

Figure 6. illustrates how adaptive RansAC combined with DBSCAN clustering finds all the lines of a complex building

roof. Edge line extraction is initiated from the farthest corner point from the centroid in the longest edge (typically has most points).

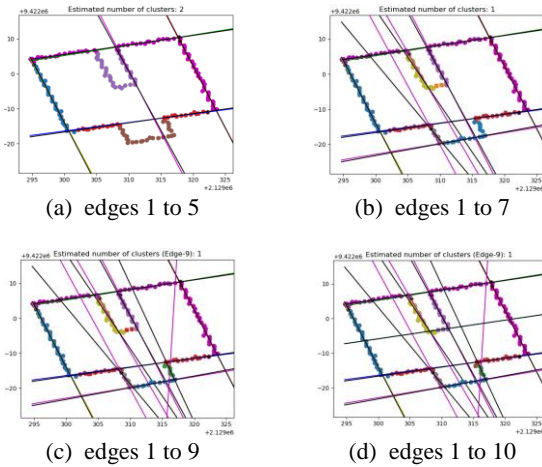


Figure 6. Line extraction from combined RansAC – DBSCAN

**3.4 DSM and true orthophotos generation**

The extracted building lines are then reprojected into 3D lines by interpolation using the original LiDAR DSM. These 3D lines are then used as hard breaklines for DSM optimization. The breaklines are expected to represent surface discontinuities and enhance the height jump at building roof edges. Figure 7. shows the difference of a DSM without breaklines (left image) and an optimized DSM with breaklines (right image). The building edge in the right image has indeed more straight representation.

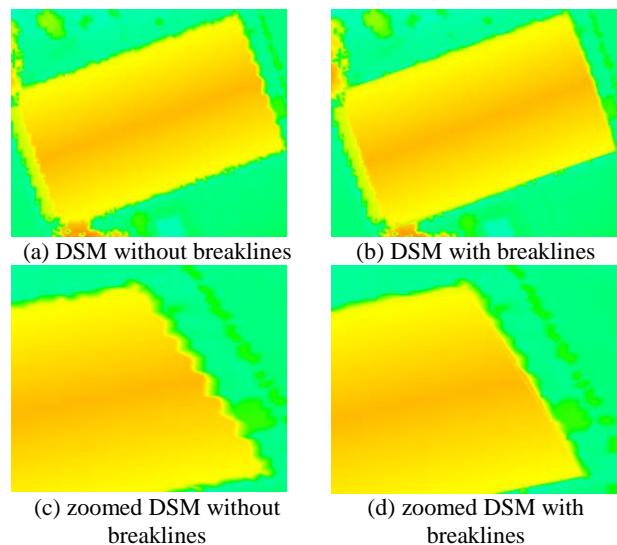


Figure 7. DSM without breaklines (left) and DSM with breaklines (right)

Based on height information, high-rise objects like buildings, trees, or bridges are set back into their true location in the true orthophotos. Therefore, a sharp edge will preserve a correct and straight building representation in true orthophotos.

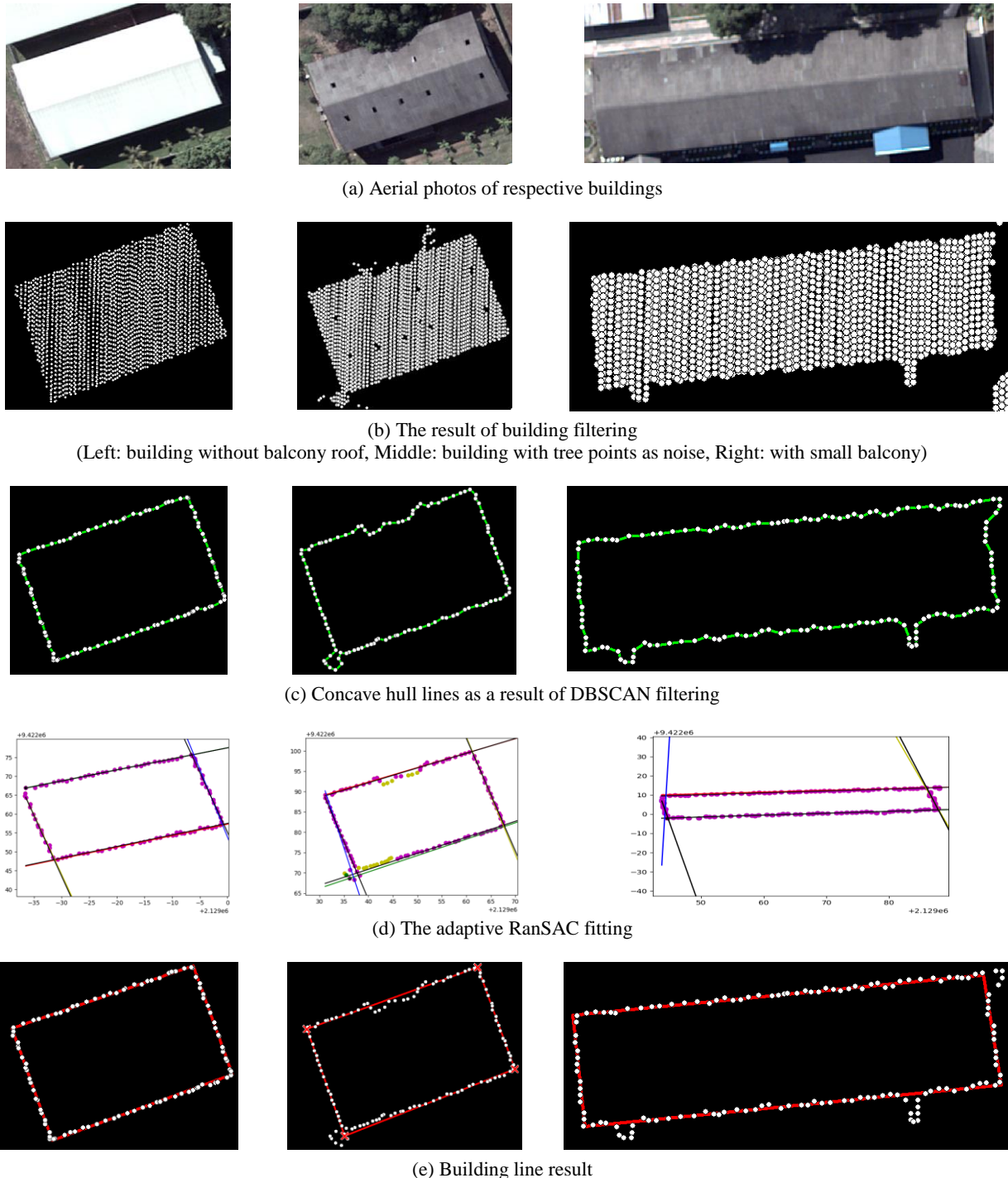


Figure 8. Illustration of the building outlines the workflow

### 3.5 Validation

The quality of the extracted building roof edges is validated based on the number of extracted buildings and the positional accuracy. It uses buildings from the base map as the reference data. Completeness of the extracted buildings resulting from our method is assessed by using the number of buildings, edges and corner points that are correctly detected. The positional accuracy is measured based on the corner points position. The squared root

of the average of the squared differences of X and Y directions is calculated to estimate the RMSE (Root Mean Square Error).

$$RMSE_x = \sqrt{\sum (X_{result} - X_{ref})^2 / n}$$

$$RMSE_y = \sqrt{\sum (Y_{result} - Y_{ref})^2 / n}$$

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

Where:

$X_{result}, Y_{result}$  = Coordinates of resulting corner points

$X_{ref}, Y_{ref}$  = Coordinates of corner points in the base map

$n$  = total number of points

We use US-NMAS (United States National Map Accuracy Standards) to determine the circular accuracy. It specifies that 90% of the well-defined points that are tested must fall within a specified tolerance. If an error is normally distributed in each the x- and y-component and error for the x-component is equal to and independent of error for the y-component, the factor 2.146 is applied to compute circular error at the 90% confidence level (Greenwalt and Schultz, 1968). Therefore, the circular map accuracy is:

$$\text{Circular Map Accuracy} = 1,5175 \times RMSEr.$$

#### 4. RESULTS AND DISCUSSION

This study focuses on the extraction of building edge extraction and particularly, aiming to improve the DSM. However, the results and discussion will focus on the results of the methodology for roof line extraction.

The line fitting procedure is successfully implemented especially for simple roof buildings. Our method is also able to produce accurate building edges in case of many outliers due to filtering errors as illustrated in Figure 8. Adaptive RanSAC succeed to find accurate lines in a simple building roof with no outliers (left), a simple building roof with many outliers (centre), and a rectangular building with outliers (right).

There are some buildings are not or incompletely detected (showed by the red arrow) as shown in Figure 9. This can be attributed to the failure of filtering in detecting building roof points. Another issue is that reason is that we found one building which has two clusters (showed by a red ellipse). The possible cause is that DBSCAN is sensitive to its input parameters, which makes it difficult to segment a point cloud of varying point density correctly.

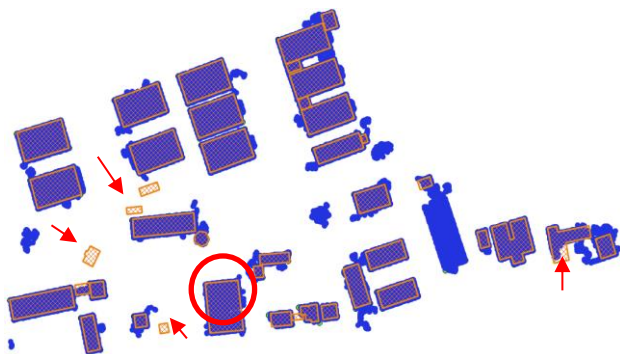


Figure 9. Base map (orange) and filtered building points (blue)

Figure 10. presents the comparison of building roof lines from the base map and roof lines resulting from the proposed method. From 42 buildings in the base map, our workflow is able to find 39 buildings. It means that the implemented workflow is able to detect more than 92% of the buildings in the study area.

Nevertheless, around 18% of the result (7 buildings) have big geometric differences:

- Five buildings are under-segmented but have a similar shape. Our roof lines result has a smaller size than the reference (showed by black arrows in Figure 10.)
- One building has a different building shape. Our line result is pentagonal while the reference has a heptagonal shape (showed by the red arrow in Figure 10.)

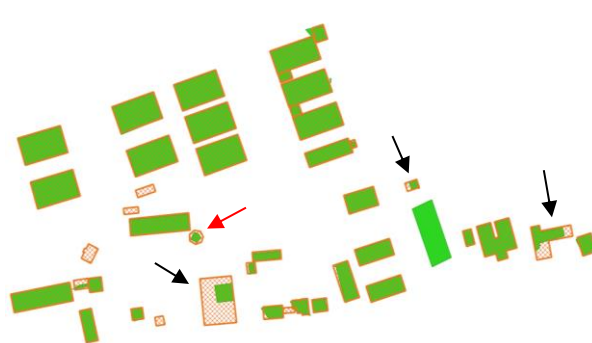


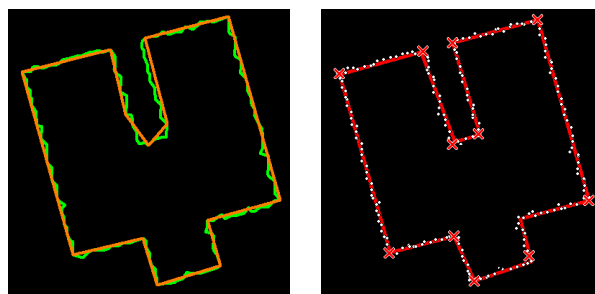
Figure 10. Base map (orange) and buildings from adaptive RanSAC (green)

We provide complete results of our building roof line extraction in Table 1.

Table 1. Completeness of Resulting Buildings

	Reference	Result
Buildings (Polygon)	42	39
Edges (Line)	235	212
Corners (Point)	235	235

Regarding positional accuracy, the average of geometric accuracy for a simple building roof is 0.576 meters. While for a successfully extracted complex building roof, it has less geometric accuracy, 0.819 meters. Still, in both cases, it has a sub-meter difference to the base map.



(a) Concave hull (green) and simplified hull lines (orange) (b) Building line result (red) of the proposed method

Figure 11. Comparison of the concave hull, a simplified concave hull, and adaptive RanSAC line

We made a comparison between building lines resulting from a concave hull line simplification and our lines. The experiment shows that a concave hull simplification may still create corners and lines that deviate from the original position in some particular part in producing a straight line, while our result is able to fit to the corner points and preserve the roof shape as shown in Figure 11a.

Due to a lack of points on the edge segments, the outlines of small buildings may deviate from its real shape or even fail to be extracted as shown in Figure 12. Another cause of errors is the presence of outliers on a small building roof. In the study area, a small balcony roof is assigned to a different segment than the main building roof. This small balcony has a size of 1m x 6 m, and it has two short edges. The boundary point distribution with very short edges (that only have one or two points) makes it impossible to find correct lines.

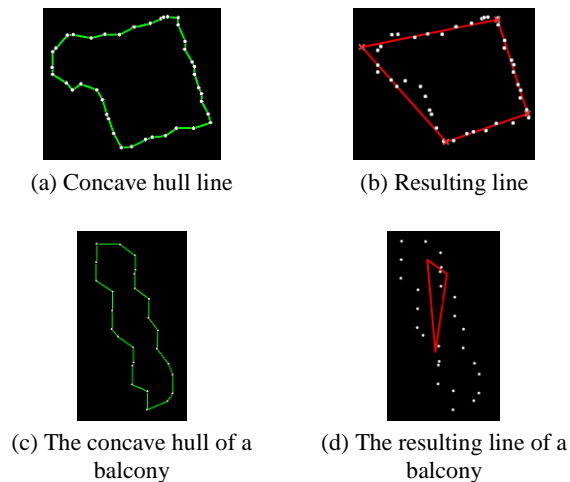


Figure 12. Incorrect results for small buildings

## 5. CONCLUSION

Our experiment shows how straight and accurate lines are extracted using the adaptive RanSAC technique. It outperforms existing methods in case of noisy edge points. The positional accuracy of the resulting lines is good, 92% buildings are detected with sub-meter accuracy, compared to the buildings in a reference base map. This workflow can be implemented for mapping activities.

However, the adapted RanSAC algorithm has difficulties on short and noisy edges. Thus, the result depends on the quality of point clouds filtering and segmentation. Rather than working per-single building, our future work will focus on how to automate the building line extraction for a complete area at once. Combining RanSAC lines from LiDAR and color information from aerial photos may give better performance and automation for building edge detection. Solutions for non-straight building edges (such as a dome roof) and buildings with shared-boundaries also need to be addressed.

## ACKNOWLEDGEMENTS

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

## REFERENCES

Asaedi, S., Didehvar, F., Mohades, A., 2013. Alpha-Concave Hull, a Generalization of Convex Hull. In: arXiv preprint arXiv:1309.7829

Awrangjeb, M., 2016. Using Point Cloud Data to Identify, Trace, and Regularize the Outlines of Buildings. *International Journal of Remote Sensing*, Vol.37, No.3, 551-579. Doi:10.1080/01431161.2015.1131868

Awrangjeb, M., Lu, G., Fraser, C., 2010. Automatic Detection of Residential Buildings using LiDAR and Multispectral Imagery. *ISPRS Journal of Photogrammetry and Remote Sensing* 65, 457-467. Doi:10.1016/j.isprsjprs.2010.06.001.

Ester, M., Kriegel, H. P., Sander, J., Xu, X., 1996. A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. *KDD-96 Proceedings*.

Federal Geographic Data Committee. 1998. Positioning Accuracy Standards, Part 3: National Standard for Spatial Data Accuracy.

Fischler, M. A., Bolles, R. C., 1981. Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. *Communications of the Acm*, Vol.24, No. 6, 726-740.

Ghosh, S., Lohani, B., 2013. Mining LiDAR Data with Spatial Clustering Algorithms. *International Journal of Remote Sensing*, 34:14, 5119-5135. Doi:10.1080/01431161.2013.787499

Greenwalt, C. R., Schultz, M. E., 1968. Principles of Error Theory and Cartographic Applications. *ACIC Technical Report No.96*.

Hu, Y., Stanley, D., Xin, Y., 2016. True Ortho generation of Urban Area using High Resolution Aerial Photos. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, Volume III-4. Prague, Czech Republic.

Kato, A., Moskal, L. M., Schiess, P., Calhoun, D., Swanson, M. E., 2010. True Orthophoto Creation through Fusion of LiDAR Derived Digital Surface Model and Aerial Photos. *IAPRS, Volume XXXVIII Part 7A, ISPRS Technical Commission VII Symposium*. Vienna, Austria.

Siddiqui, F. U., Teng, S. W., Awrangjeb, M., Lu, G., 2016. A Robust Gradient Based Method for Building Extraction from LiDAR and Photogrammetric Imagery. *Sensors*, 16(7), 1110. Doi:10.3390/s16071110.

Wang, Q., Yan, L., Sun, Y., Cui, X., Mortimer, H., Li, Y., 2018. True Orthophoto Generation using Line Segments Matches. *The Photogrammetric Record*, Vol. 33, Issue 161.

Wu, J., Jie, S., Yao, W., Stilla, U., 2011. Building Boundary Improvement for True Orthophoto Generation by Fusing Airborne LiDAR Data. *IEEE, Joint Urban Remote Sensing Event*. Munich, Germany.

Yang, B., Fang, L., Li, J., 2013. Semi-Automated Extraction and Delineation of 3D Roads of Street Scene From Mobile Laser Scanning Point Clouds. *ISPRS Journal of Photogrammetry and Remote Sensing* 79, 80-93. Doi:10.1016/j.isprsjprs.2013.01.016.

Zhao, Z., Duan, Y., Zhang, Y., Cao, R., 2016. Extracting Buildings from and Regularizing Boundaries in Airborne LiDAR Data using Connected Operators. *International Journal of Remote Sensing*, Vol. 37, No.4, 889-912. Doi:10.1080/014311611.2015.1137647.