

## Article

# A Simple Yet Effective Approach of Building Footprint Extraction in Indonesia

Danang Budi Susetyo <sup>1,\*</sup>, Aldino Rizaldy <sup>1</sup>, M. Irwan Hariyono <sup>1</sup>, Nugroho Purwono <sup>1</sup>, Fahrul Hidayat <sup>1</sup>, Rizka Windiastuti <sup>1</sup>, Tia Rizka N. Rachma <sup>1</sup> and Prayudha Hartanto <sup>1</sup>

<sup>1</sup> Geospatial Information Agency, Cibinong, Bogor, Indonesia 16911

\* Correspondence: danang.budi@big.go.id

**Abstract:** Topographic mapping using stereo plotting is not effective because it takes much time and labor-intensive. Thus, this research was conducted to find the effective way to extract building footprint for mapping acceleration. Building extraction method in this process comprises four steps: ground / non-ground filtering, building classification, segmentation, and building extraction. Non-ground points from filtering process were classified as building with the algorithm based on multi-scale local dimensionality to separate points at the maximum separability plane. Segmentation using segment growing was used to separate each building, so edge detection could be conducted for each segment to create boundary of each building. Lastly, building extraction was conducted through three steps: edge points detection, building delineation, and building regularization. With 10 samples and step 0.5, classification resulted quality and miss factor of 0.597 and 0.524, respectively. The quality was improved by segmentation process to 0.604, while miss factor was getting worse to 0.561. Meanwhile, on average shape index value from extracted building had 0.02 difference and the number of errors was 30% for line segment comparison. Regarding positional accuracy using centroid accuracy assessment, this method could produce RMSE of 1.169 meters.

**Keywords:** building footprint; LiDAR; classification; segmentation

## 1. Introduction

The availability of high-quality geospatial data, including building footprint as part of digital topographic map, is necessary for bigger purposes, including 3D city models [1], [2] and spatial planning. In Indonesia, spatial planning plays more concern due to limited availability of topographic data that inhibits spatial planning in many regions because spatial planning in Indonesia requires topographic map as a reference for base map.

Traditionally, topographic data production uses stereo plotting process using photogrammetry technique. However, manual stereo plotting takes much time and labor-intensive, so it is not effective to be applied for large scale topographic mapping in the whole area of Indonesia. Due to high needs for large scale topographic maps, acceleration of large-scale mapping must be conducted, therefore innovation in large scale mapping method should be discovered.

Automation of map feature extraction is one of the solutions for mapping acceleration. The automation will reduce the amount of time and human resources, so that it will increase the speed of topographic map production. One of the map features that has been evaluated to be extracted in this research is building. The need of the highly accurate building footprint can be fulfilled by manual digitization, but it can be very time consuming. Compared to the other map features, building is one of the most complex features. It is not easy to develop the algorithm for automatic building extraction. Moreover, most buildings in Indonesia are very dense and close to each other, and many of them are in irregular shapes, making it a more challenging research.

Research of automatic building extraction is not something new, many methods have been developed effectively and accurately. Various base data from aerial photo, LiDAR, satellite imagery, or combination of them has been examined in order to find the best method for building extraction. A recent technology is based on deep learning algorithms, and many researches about that for building extraction have been conducted, for example using neural networks and Markov Random Fields (MRF) [3], end-to-end trainable gated residual refinement network (GRRNet) [4], Fully Convolutional Network [5], Convolutional Neural Network [6], Deep Convolutional Network [7] [8], or Deep Fully Convolutional Networks [9]. Another method such as supervised and unsupervised segmentation are still used and developed, for instance by [10] in which it compared supervised and unsupervised multiresolution segmentation for building extraction.

The development of LiDAR creates a new approach for automatic map features extraction. LiDAR is not only used for DEM extraction, but also for detecting planimetric features such as buildings and roads. Commonly, building extraction from LiDAR starts from point cloud classification of building class (or building roof class). Many algorithms have been developed for building class classification from LiDAR point clouds, for instance using input attributes from DSM and ortho image [11], a supervised parametric classification algorithm [12], the theory of Dempster-Shafer for two different stages of classification process [13], the geometry and echo information of LiDAR point cloud to estimate parameters for the learning model [14], and a multiple-entity based classification method [15]. Regarding building extraction, a lot of research found alternative methods for building extraction from LiDAR, for example using OBIA (Object Based Image Analysis) with the use of class modelling methods [16], data fusion of point and grid-based features using a graph cuts algorithm [17], a novel ordered points-aided Hough Transform (OHT) [18], a modified LEGION segmentation [19], automatic process with emphasis on top-down approaches [20], and rule-based segmentation of non-ground LiDAR point clouds [21].

Combination and integration using both raster and point based data is another option. [22] developed automatic building detection algorithm using LiDAR data and multispectral imagery, combining masks creation from LiDAR to extract line segments and NDVI from imagery to remove line segments around trees. [23] proposed a method for automatic 3D roof extraction through an integration of LiDAR and multispectral imagery, in which segmented non-ground points were used to extract the roof planes, while the classified image as roof edge and roof ridge were used as baselines to locate the nearby LiDAR points of the neighboring planes to extract the complete roof plane. [24] integrated LiDAR and optical multi-view images by incorporating the segmented roof points and 2D lines extracted from aerial images for reconstructing 3D roof models.

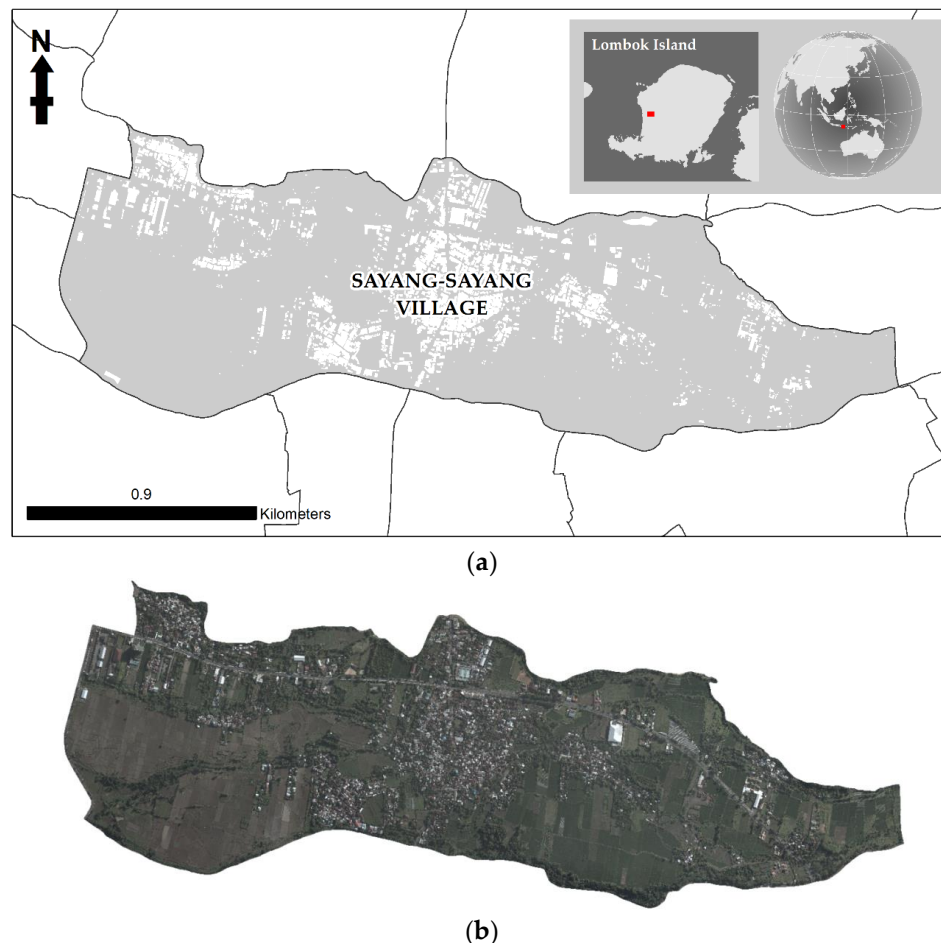
Combination of raster image and LiDAR point cloud data is not only applicable for detecting building footprint, but also good for 3D building model extraction. 3D modeling both from images and LiDAR has been an active research area in the photogrammetry, computer vision, and computer graphic communities [25]. 3D building modeling from LiDAR has been developed, for instance, by [26] that developed reconstruction approach to produce 3D building model with LOD2, [27] that built new framework to create compact building models automatically, or [28] that proposed fast and automatic method to create 3D watertight building models. Specifically using Indonesian data, [29] built building models using plane fitting to sets of non-ground points. Similar to building footprint extraction, combination of raster and point clouds data is also possible to generate 3D building models. For example, by integrating multi-view aerial imageries and LiDAR data to reconstruct 3D building models [24] or integrating multi-source point clouds and oblique remote sensing imageries for accurate reconstruction of the LoD3 building models [30]. Thus, utilization of LiDAR or combination of LiDAR and imagery has been used beyond building footprint extraction, namely for 3D building modeling, or even city modeling.

The purpose of this research was to find the best approach to extract building footprint automatically for topographic mapping acceleration in Indonesia. Although this research did not use recent technologies such as deep learning or machine learning, it was effective enough to create

accurate building geometries. Moreover, what is most needed in Indonesia is finding a simple but effective approach to extract topographic map features, since completing topographic map for a large country such as Indonesia is still challenging. This research is expected to improve the business process of topographic mapping in Indonesia to be cheaper, faster, and more effective.

## 2. Materials and Methods

This research used LiDAR data acquired in 2016 (for building footprint extraction) and topographic map of Indonesia/RBI5K mapped in 2017 (as a reference dataset) in Lombok region, Nusa Tenggara province, Indonesia. More specifically, the research was conducted at Sayang-sayang village, located in Cakranegara district in Mataram City, Lombok (Figure 1). This area was chosen because it represented various characteristic of buildings, containing both dense and sparse building groups, so that the analysis could be conducted based on the density of buildings. Regarding the specification of the data, the density of point clouds was 4 ppm (points per meter square) or higher, meaning that for each meter square area there were at least four points. This specification was suitable for Indonesia's topographic mapping at 1:5,000 map scale.



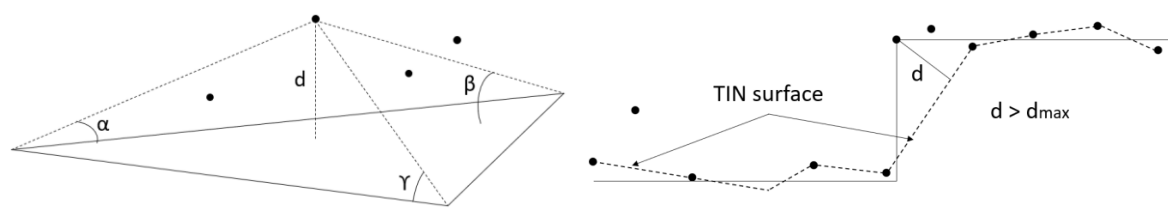
**Figure 1.** (a) Area of interest, located in Sayang-sayang village, Nusa Tenggara Barat, Indonesia; (b) Aerial photo of research area (acquired in 2016), cropped using village boundary from topographic map.

### 2.1. Ground / Non-ground Filtering

Ground / non-ground filtering was one of the important steps in this research, because it would affect building classification result. Ground and non-ground classification was not only useful for DTM extraction, but also for identifying building because it could reduce number of points [31]. In

that sense, the next step of point cloud classification into vegetation and building classes was conducted in non-ground points only. Therefore, point cloud separation to ground and non-ground classes determined the quality of building classification.

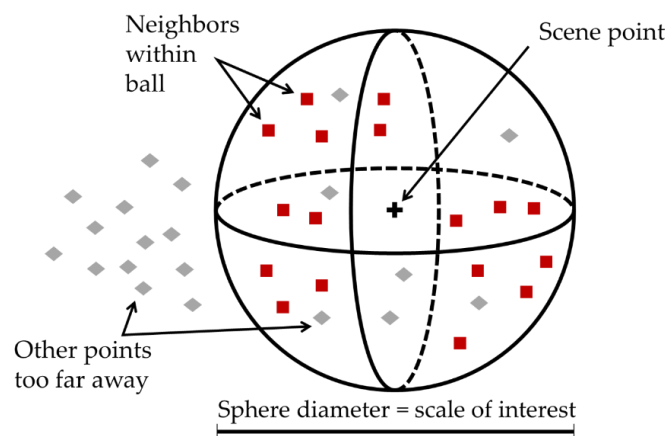
Classification of ground and non-ground classes was performed using Adaptive-TIN Surfaces algorithm (Figure 2) in Terrasolid software. This algorithm contains four steps, namely: calculating initial parameters using all data, selecting seed points, iterative densification of the TIN, and continuing until all points were classified as ground or object [32]. The inputs such as maximal building size, terrain angle, iteration angle, and iteration distance were crucial, in which the values used were 100 meters for maximal building size, 88.0 degrees for terrain angle, 5.00 degrees to plane for iteration angle, and 1.40 meters for iteration distance. Maximum building size shows area that has at least one initial ground point, terrain angle assumes the steepest angle between points in generated TIN, terrain angle depends on terrain condition in which smaller iteration angle is better for flat terrain and vice versa, and iteration distance is maximum distance between point to triangle [33].



**Figure 2.** Parameters of TIN densification in Adaptive TIN Models algorithm [32].

## 2.2. Building Roof Classification

Building roof classification used the algorithm developed by [34]. It was based on a multi-scale local dimensionality to separate points into two classes at the maximum separability plane. The classification considered how the cloud geometrically looked like at a ball geometry at a given scale, then the best combination of scales at which dimensionality was measured. The diameter of the ball was defined as a scale while point of interest was located at the center, as shown in Figure 3.



**Figure 3.** Neighborhood ball [34].

To determine whether the cloud appeared locally at a given scale in 1D, 2D, or 3D, the proportion of variance by the eigenvalues resulting from Principal Component Analysis (PCA) was used, with the equation (1) defined the proportion of variance [34].

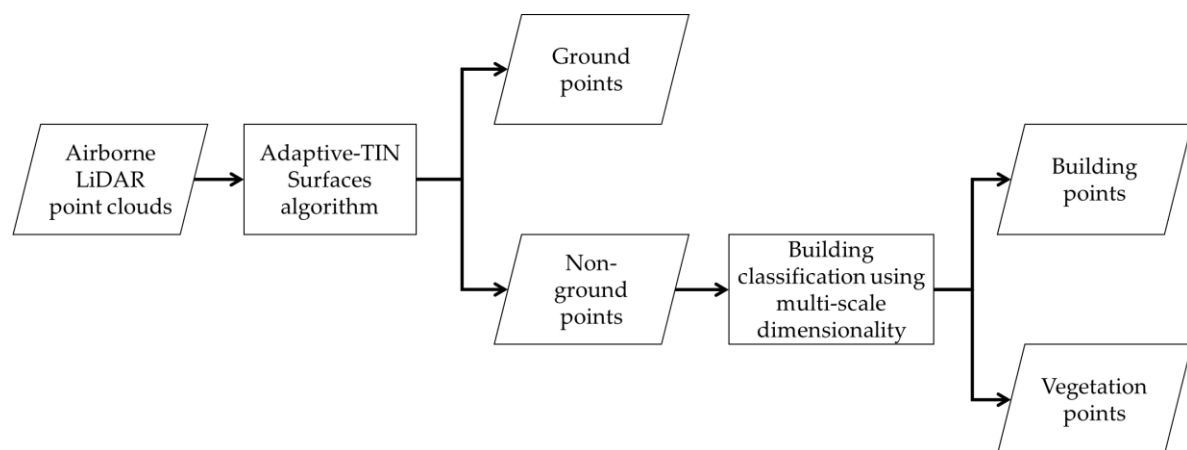
$$pi = \frac{\lambda_1}{\lambda_1 + \lambda_2 + \lambda_3} \quad (1)$$

where if  $\lambda_i, i = 1 \dots 3$  be the eigenvalues, ordered by decreasing magnitude:  $\lambda_1 \geq \lambda_2 \geq \lambda_3$ . The number of eigenvalues that accounted for the total variance in the neighborhood ball determined whether the points were distributed in one, two, or three dimensional around the reference scene point.

The algorithm used a linear Support Vector Machines (SVM) created by CANUPO training plugin in Cloud Compare software. A linear classifier proposed one solution in the form of hyperplane that best discriminated and separated two classes (e.g. vegetation against building). It used weight vector ( $w$ ) and bias ( $b$ ), in which the hyperplane was defined by equation  $w^T X - b = 0$ .  $w$  and  $b$  were used to project the feature space on the hyperplane, then the distance to the hyperplane  $d_1 = w_1^T X - b_1$  was calculated for each point. The second distance  $d_2$  was obtained by repeating the process in order to get the second-best direction orthogonal to the first, then both distance  $d_1, d_2$  were used as coordinates at 2D plane of maximal separability.

The plane of maximal class separability only kept two main components, so in this research point clouds were separated into two categories: building and vegetation. To do this classification step, we first created training area that represented both categories. Then, some parameter numbers were decided to create this training data, including minimum and maximum scale to be limits of multiscale dimensionality. The unit of the scale referred to the unit of the point cloud. In this research, the chosen number for scales was 2-10 because the buildings length in the research area were considered between 2 and 10 meters since they were located at settlement area. Another parameter was step, defined as the distance unit from the minimum to the maximum scale, in order to achieve multi-scale features. Small step means more features, otherwise, larger step means less features. In this research, the chosen number for step was 0.5 meters due to the density of LiDAR point clouds is 4 ppm (points per meter square).

Shortly, the whole process for all classification steps is explained in Figure 4. It started from ground and non-ground classification, then non-ground points were classified as building and vegetation points. All processes here were binary classification. The building points were then used in the next stage, namely segmentation.



**Figure 4.** Classification process from unclassified point clouds to classified building points.

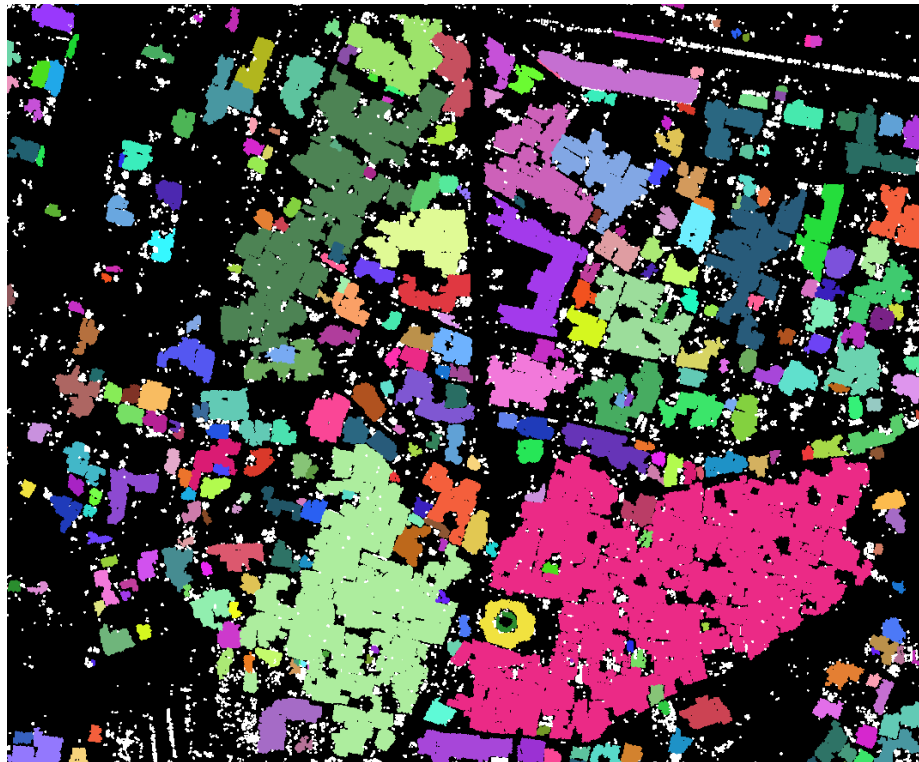
### 2.3. Segmentation

Segmentation in point cloud is used to distinguish group of points based on feature similarity. In order to improve classification result, [35] used segment-based features rather than traditional point-based features for point based classification. The segment-based features led to more accurate classification result. In our research, segmentation was used to separate each building, so that edge

detection could be conducted per segment to create boundary of each building. The data used in this stage was the previous classification result, namely classified building points.

Segment growing [35] algorithm was chosen for segmentation process. We used Point Cloud Mapper software for the implementation of the segment growing segmentation. In this algorithm, the test for accepting neighboring points as extensions of a segment was based on similarity of feature values and normal vectors scaled by the planarity to distinguish between surfaces with different orientations [35]. Seed points with similar feature values would be extended and merged with their neighbor points if their values were close to the average feature values for each segment. For instance, [36] used such segment growing based on echo widths of full waveforms to separate vegetation from non-vegetation. Combination with the normal vector direction scaled by planarity was used to distinguish between surfaces with different orientations, in which when  $\lambda_1 > \lambda_2 > \lambda_3$  were the eigenvalues of the co-variance matrix, the planarity could be expressed as  $\frac{(\lambda_2 - \lambda_3)}{\lambda_2}$  [35].

Segmentation was also used to remove small segments and unsegmented points, therefore it eliminated building errors in the building extraction process. Unsegmented points, colored as white points (Figure 5) and did not have segment number, were points without sufficient nearby similar points to start a new seed [35].



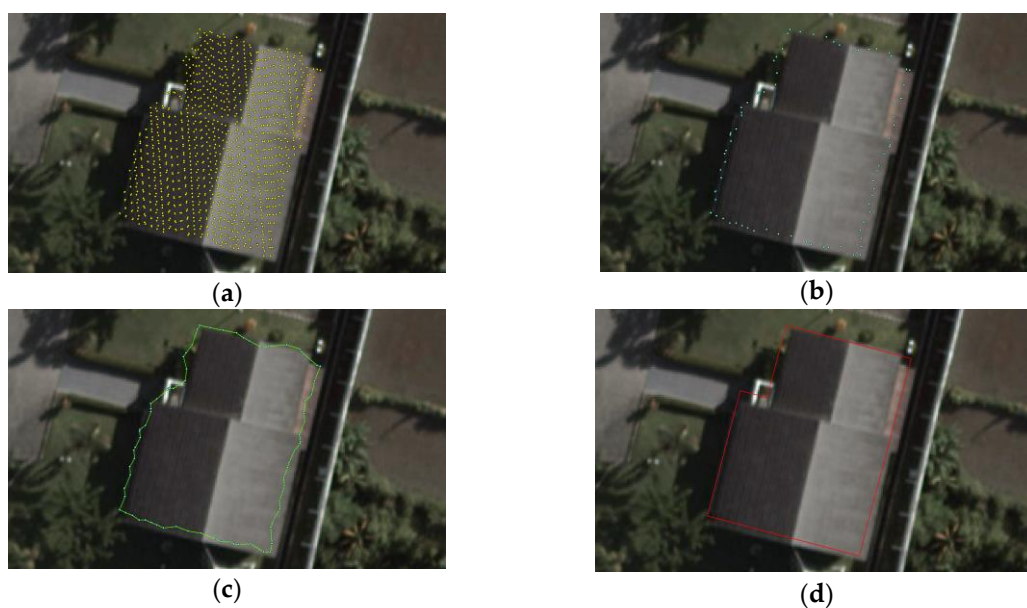
**Figure 5.** Unsegmented points (colored as white points) around segmented point clouds.

#### 2.4. Building Footprint Extraction

Building extraction process started from boundary detection for each segment to create delineation of coarse building footprint from those edge points. After segmented points were obtained using segment growing, then edge points of each segment were extracted using developed boundary package in MATLAB before delineation of coarse building boundaries are created from those edge points (Figure 6a, 6b, and 6c). The algorithm of boundary points detection returned points that represented boundary arounds the building roof points, and unlike the convex hull, the boundary could shrink towards the interior of the hull to envelop the points (<https://www.mathworks.com/help/matlab/ref/boundary.html>). Then, points for each segment were

connected with lines using Points to Line tool so it produced raw building footprints. Lastly, to produce smooth and sharp building footprint, regularization of building boundary was conducted using Building Regularization tool that applied algorithm developed by [37] (Figure 6d). The implementation of the edge delineation and building footprint regularization was performed in ArcGIS Pro software (<https://www.esri.com/en-us/arcgis/products/arcgis-pro/resources>).

In order to achieve good geometry of building regularization, reconstruction of coarse building required support  $90^\circ$  and  $45^\circ$ . This algorithm was also suitable to arcs, in which the arc passing through considered locations differed from the segment by the necessity to define the radius [37]. Building regularization required the setting of tolerance parameter that described the smallest segment length in the output features, and it was set to 1 meter. This number was based on experiments using several numbers, considering the difference between the number of segments from extracted and reference segments, so it could be assumed that extracted objects had similar level of detail to reference objects.

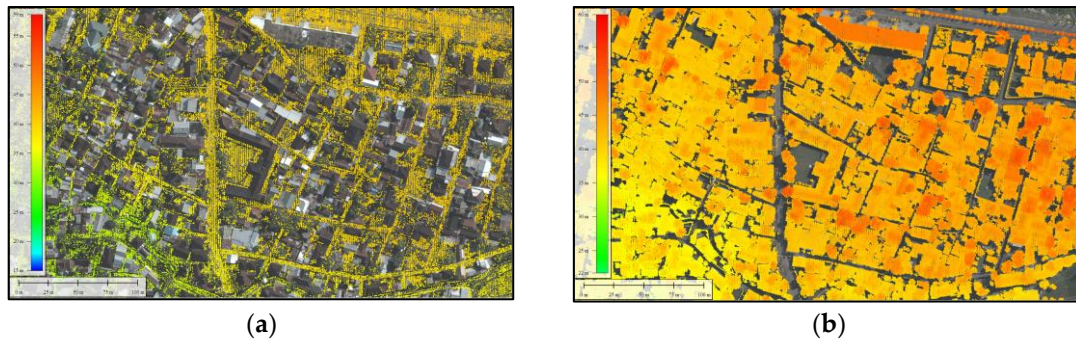


**Figure 6.** (a) Segmented points cloud from segment growing helped to identify each single building; (b) From segmented point cloud, edge points were created using developed boundary package from MATLAB; (c) Edge points were used to delineate coarse building boundary, so coarse boundaries for each building were created; (d) Regularization process to coarse boundary was needed to produce the final building footprint.

### 3. Results and Discussion

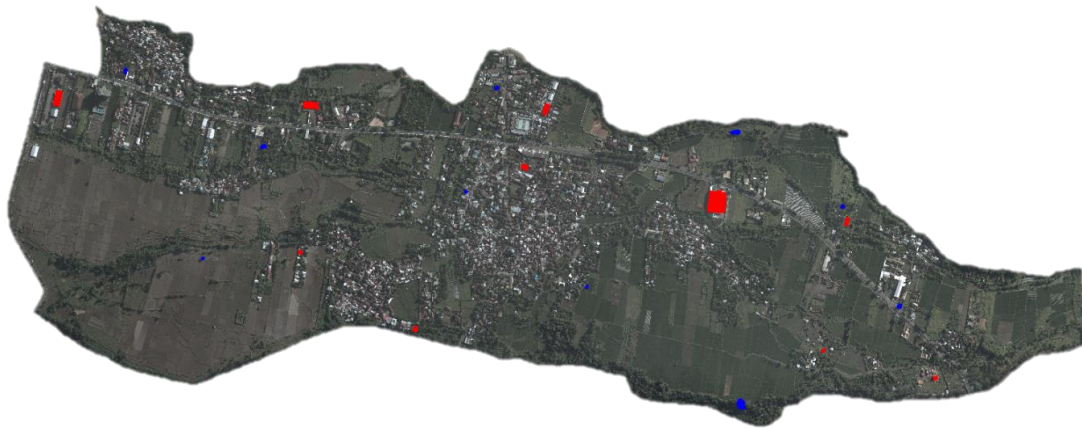
#### 3.1. Classification Result

Figure 7 shows the result of filtering process, color-coded by the elevation. Figure 7a shows all points in the research area, while 7b depicts the result of ground and non-ground filtering process, with background of ortho photo. As can be seen in Figure 7b, ground points were removed, as illustrated in paddy field area, because it was an open area and no building existed in that area, almost all points there were removed.



**Figure 7.** The result of ground and non-ground classification: (a) Ground points; (b) Non-ground points, consist of building and vegetation.

Then, analysis of building classification was conducted. Figure 8 shows the distribution of training area for both categories, namely building and vegetation. Red color indicates building area, while blue color shows vegetation area. Each category had 10 objects as training area, which was obtained from manual classification by human interpretation and well distributed in the whole research area. It should be noted that even though the number of samples were small, the classification performed well as can be seen in the result.



**Figure 8.** Training area for each category (building and vegetation). Red points are building, while blue points are vegetation.

Classification result of training data was shown in a plane of maximal separability and separation line. Balanced accuracy ( $ba$ ) and Fisher Discriminant Ratio ( $fdr$ ) was used to quantify the performance of the classifier, where a large  $ba$  value indicated a good recognition rate, while a large  $fdr$  value indicated that classes were well separated, and obtained using these equations [34]:

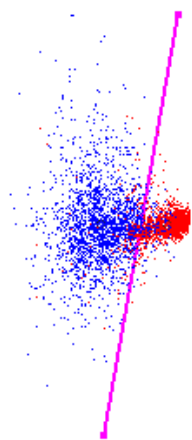
$$ba = \frac{1}{2}(a_v + a_b) \quad (2)$$

- $a_v = \frac{tv}{tv+fb}$
- $a_b = \frac{tb}{tb+fv}$
- $tv, tb, fv, fb$  = the number of points truly( $t$ ) / falsely( $f$ ) classified into the vegetation( $v$ )/building( $b$ ) classes

$$fdr = (\mu_2 - \mu_1)^2 / (v_1 + v_2)$$

(3)

- $\mu_c$  and  $v_c$  = the mean and variance of the signed distance  $d$  for each class  $c$



**Figure 9.** Classifier definition in the plane of maximal separability. Blue: vegetation samples, red: building samples. From statistic calculation, the value of *ba* was 0.916, while the value of *fdr* was 3.811.

The statistic showed that *ba* value gave very good result, reaching 91.6%, meaning that the algorithm could recognize both building and vegetation well (Figure 9). However, *fdr* value did not indicate good separation of those classes because it was only 3.811. To test the effect of number of samples and step parameter to *ba* and *fdr* results, experiments with various parameters were conducted. We used several numbers of sample from 10, 15, 20, 25, and 30, and also changed the step parameter by 0.5 and 1, where the result is shown in Table 1. It can be seen that addition of samples and changing step parameters did not improve the results of *ba* and *fdr* too much. In conclusion, it was still difficult to increase *fdr* value to reach more than 10 as produced by [34]. Unlike our work that used airborne laser scanning, [33] used much higher point density of terrestrial laser scanning. Therefore, it was expected that our work achieved lower *fdr* value than [33].

**Table 1.** The result of *ba* and *fdr* value with various number of samples and step parameters.

Number of Samples	Step	<i>ba</i>	<i>fdr</i>
10	0.5	0.916	3.811
	1	0.912	3.414
15	0.5	0.926	3.651
	1	0.923	3.763
20	0.5	0.918	3.583
	1	0.909	3.213
25	0.5	0.92	3.463
	1	0.909	3.218
30	0.5	0.916	3.344
	1	0.914	3.203

Besides comparing the *ba* and *fdr* value, analysis of classification accuracy was conducted using matched rates analysis, consisting of completeness, correctness, quality, and miss factor, defined by equation (4) – (7) [38]. Initially, those equations were used to assess extracted buildings instead of classified building points, but in this research, it was adapted to examine the accuracy of building classification result. Reference points here were non-ground points cloud that overlapped with

building geometry from topographic map, which were obtained from manual stereo plotting. It was assumed that those points were the correct building points and there was no geometrically offset between point cloud and topographic map.

$$\text{Completeness} = \frac{TP}{TP+FN} \quad (4)$$

$$\text{Correctness} = \frac{TP}{TP+FP} \quad (5)$$

$$\text{Quality} = \frac{TP}{TP+FN+FP} \quad (6)$$

$$\text{Miss factor} = \frac{FN}{TP} \quad (7)$$

- TP (true positive) = the common points of classified points and reference points
- FP (false positive) = the classified points but not reference points
- FN (false negative) = the points that belong to the reference but not the classified points
- Completeness = the percent of classified points being correctly detected with respect to the reference data
- Correctness = the percent of correctly detected part to the classified points
- Quality = metric that combines the completeness and correctness
- Miss factor = the rate of missed building points

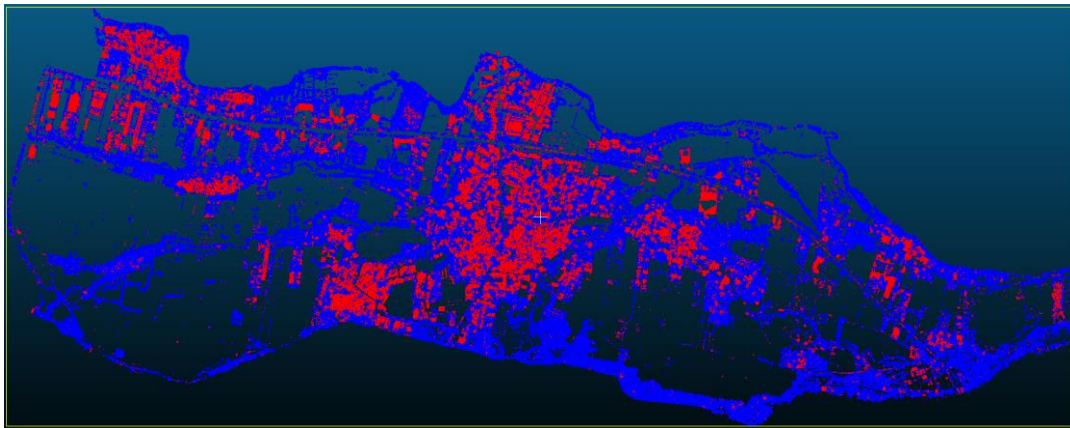
The result of matched rates analysis is shown in Table 2. With 10 samples and step 0.5, it produced the highest correctness at once the lowest completeness and quality compared to the other parameters. The best result was produced using 25 samples and step 1, with quality of 0.666 and miss factor of 0.263. However, since the purpose of this research was to find the simple and effective method for building extraction, the least number of samples was chosen. Meanwhile, the step used was 0.5, because when the specification of point density was 4 ppm, it was assumed that the distance between 2 points was at least 0.5 meter.

**Table 2.** The result of matched rates analysis.

Number of Samples	Step	Completeness	Correctness	Quality	Miss Factor
10	0.5	0.656	0.869	0.597	0.524
	1	0.719	0.846	0.635	0.392
15	0.5	0.695	0.849	0.619	0.438
	1	0.740	0.831	0.643	0.352
20	0.5	0.787	0.802	0.659	0.270
	1	0.792	0.800	0.661	0.263
25	0.5	0.790	0.806	0.664	0.266
	1	0.792	0.807	0.666	0.263
30	0.5	0.773	0.822	0.663	0.293
	1	0.769	0.826	0.662	0.300

In the next step, the training data was applied to classify the whole research area. This area was covered by buildings (with various characteristic such as big, small, regular, and irregular), vegetation with variability of canopy diameter, and also open area like paddy field and roads.

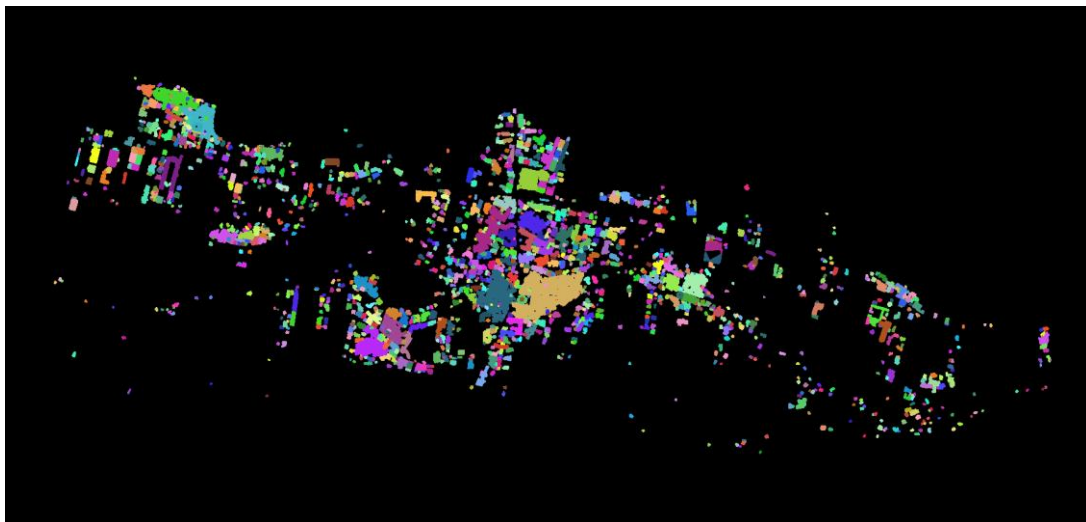
Generally, the buildings had similar shape and geometry because most of them were settlement and housing area. Figure 10 shows the classification result.



**Figure 10.** Building and vegetation classification result. Red color indicated building class, while the blue one was vegetation class.

### 3.2. Segmentation Result

Figure 11 shows segmentation result from Point Cloud Mapper software, in which each color indicated different segment. Segment growing result showed point clouds were segmented to each building, so that segmentation result could be identified as building separation. This process was also used to remove small segments and unsegmented points, so small and unidentified objects that classified as errors were eliminated here.

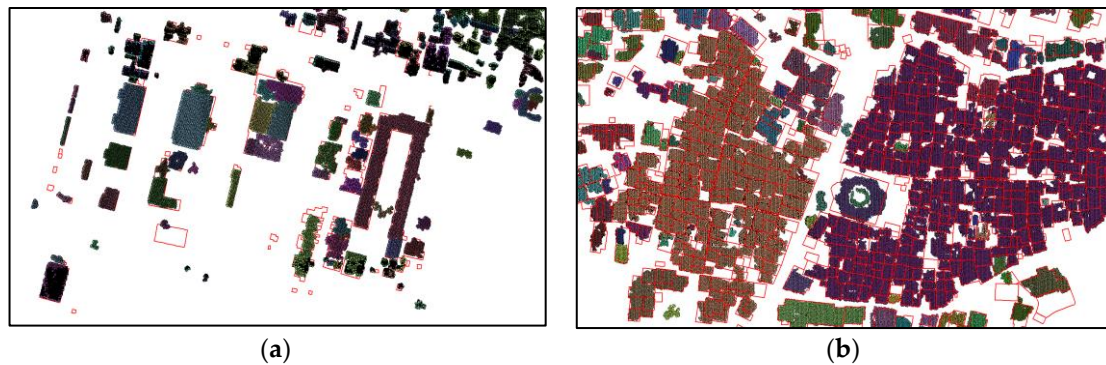


**Figure 11.** Segmentation result using segment growing algorithm.

Compared to classification result, segmentation increased the quality of building points based on assessment method in the previous chapter. The quality was escalated from 0.597 to 0.604, while correctness was improved from 0.869 to 0.913. Nevertheless, the completeness and miss factor got worse, where completeness was reduced from 0.656 to 0.641, while miss factor was increased from 0.524 to 0.561. It was because there were more points eliminated in this stage, therefore the percentage of missing objects was higher.

Building separation from segmentation process was compared to building polygon from topographic map at 1:5,000 scale. In general, segmentation could produce good result in sparse buildings (Figure 12a), but the error of segmentation result was bigger when the buildings were

denser, because many of buildings that were very close to each other were segmented as one segment (Figure 12b).



**Figure 12.** (a) Segmentation result in sparse building area; (b) Segmentation result in dense building area.

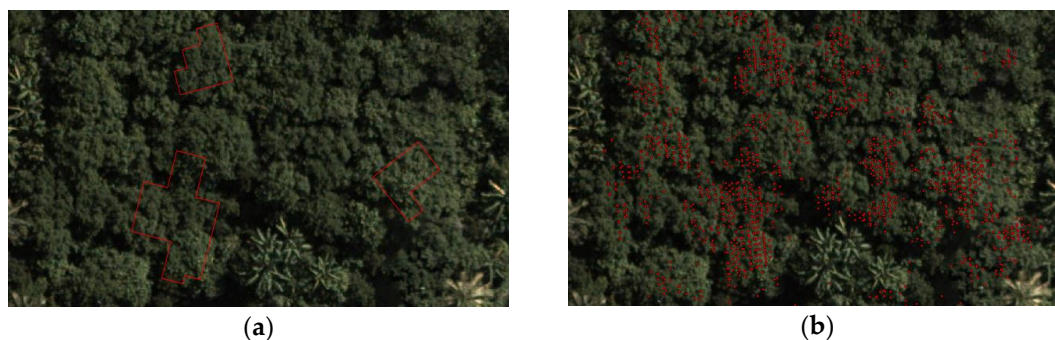
### 3.3. Building Extraction Result

#### 3.3.1. Visual Analysis

Visual analysis was conducted by comparing extracted buildings and ortho photo. In general, the errors from visual analysis can be divided into five categories:

1. Extracted buildings were generated in non-building area (visually checked on ortho photo)

Figure 13a shows the example of this category, where extracted buildings were located in non-building area. Based on the ortho photo, it was known that the object was vegetated area. It was caused by error of classification result, where some non-ground points were classified as building in non-building area (Figure 13b). This error may be corrected by changing the classification parameters.

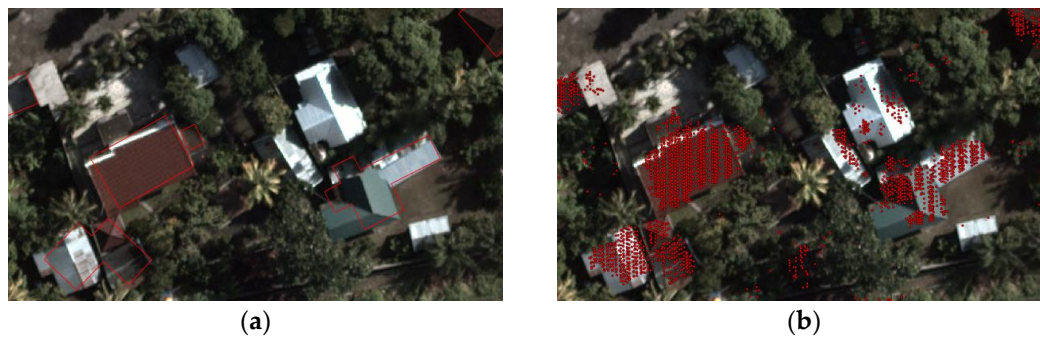


**Figure 13.** Error where extracted buildings were generated in non-building area based on ortho photo; (a) Error in extracted building footprints; (b) Error in building classification of non-ground point clouds.

2. Extracted buildings were not generated in building area (visually checked on ortho photo)

This category is opposite to the previous category, where the true buildings based on ortho photo were not generated in extracted building. It can be seen in Figure 14a that some of buildings were not generated. Figure 14b shows those buildings only had a small number of points so it was not possible to create building boundary from them. Such points were also possible to be removed during segmentation process, so it increased the possibility for them to be not extracted in the building delineation. It may be corrected by changing the classification parameters. This

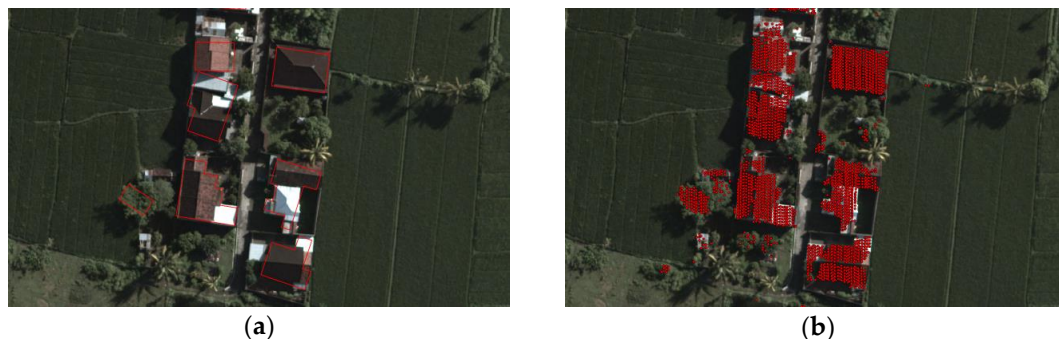
error would need further effort to delineate building boundary using another method like manual digitization.



**Figure 14.** Error where extracted buildings were not generated in building area based on ortho photo; (a) Error in extracted building footprints; (b) Error in building classification of non-ground point clouds.

3. Extracted buildings had different orientation from true building (visually checked on ortho photo)

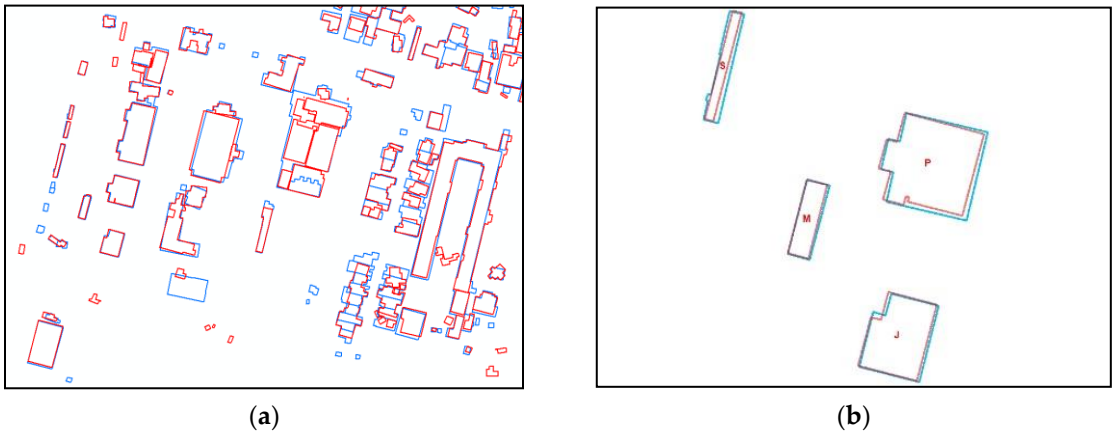
Another error was when extracted buildings had different orientation compared to the true building that was seen from ortho photo. Even though the possibility of this error was much less than the previous errors, it was still possible, as depicted in Figure 15a. This kind of error could be caused by building regularization algorithm that assumed building geometry has  $90^\circ$  or  $45^\circ$  angle, or error during classification and segmentation. One of the solutions was by improving segmentation result, but in this research, it had not been conducted yet.



**Figure 15.** Error where extracted buildings had different orientation from true building based on ortho photo; (a) Error in extracted building footprints; (b) Error in building classification of non-ground point clouds.

### 3.3.2. Statistical Analysis

A number of related data samples were taken to assess the result of building extraction from LiDAR. Figure 16 illustrates the result of building extraction which depicted as building footprints, where the red line polygon was building feature from the LiDAR and the blue one was the feature from reference dataset. It can be seen that the result of building extraction had a similar pattern to the reference map. However, some errors were found, i.e. the building that was separated into several buildings. In addition, several small buildings were missing and not well extracted, and vice versa. Also, several building footprints were extracted falsely in non-building positions.



**Figure 16.** (a) The sresult of building extraction; (b) Samples of extracted building.

Three methods were used to assess the results of geometry of footprint building extraction, which were segment of line comparison, shape index similarity [39]–[41] and edge point-correspondence [39], [42]. The segment of line comparison was used to evaluate the number of sections of every line resulted from samples against reference dataset. Statistically, both data were calculated to generate distinction of the line segments and counted it into a percentage which indicated the level of error. Matrix of the line segment comparison is shown as Table 3. Shape index similarity was used to enumerate the shape ratio between two objects. This method expressed a ratio of area to the length of the circumference [40]. Each sample of extracted building was calculated as a polygon, where the shape index was calculated between perimeter in giving an internal area of a polygon. If the shape index was 1, it meant the polygon calculated was circular, while the higher number of 1 indicated the bar-shaped of the polygon. If the polygon was square, it had an index of approximately 1.13. It is possible to express the shape index equation as follows:

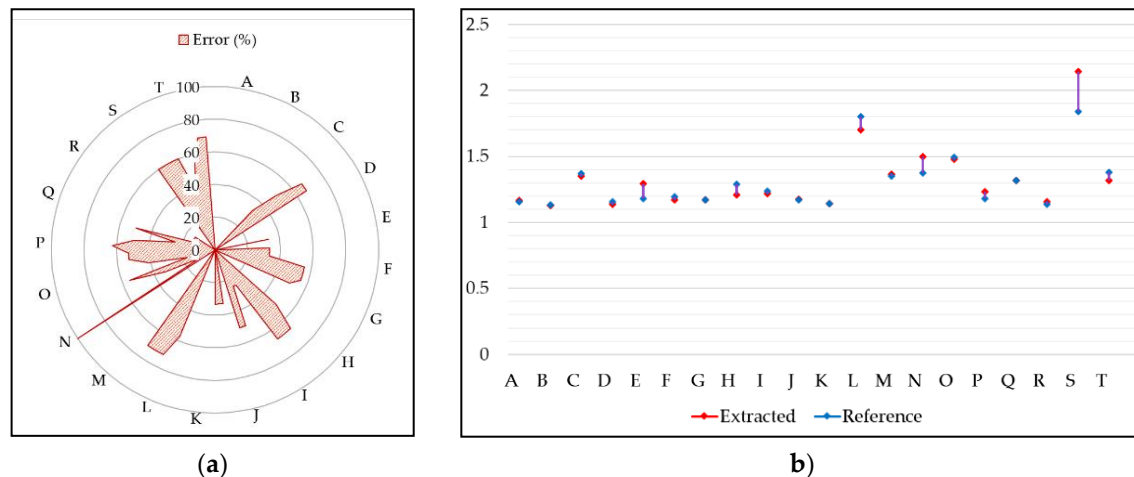
$$Di = \frac{pi}{2\sqrt{A\pi}} \tag{8}$$

where:  
 $Di$  : index  
 $pi$  : perimeter (m)  
 $A$  : area (m<sup>2</sup>)

**Table 3.** Comparison of segment number between samples of reference datasets and building extraction data.

A				B				C				D				E			
4				4				12				4				6			
4	4	4	4	4	4	4	4	8	6	4	4	4	4	4	4	8	6	4	4
F				G				H				I				J			
9				4				12				8				6			
6	4	4	4	6	4	4	4	6	4	4	4	10	6	4	4	6	4	4	4
K				L				M				N				O			
4				14				4				9				17			
4	4	4	4	6	4	4	4	4	4	4	4	18	10	6	4	14	10	8	8
P				Q				R				S				T			
8				7				4				10				13			
13	12	6	4	8	6	6	6	4	4	4	4	4	4	4	4	8	6	4	4

Table 3 shows the number of line segment in each data sample, where it was arranged downward respectively in sample ID, number of segments of reference datasets, and number of segments of extracting data. Each number of segments of extracting data is shown as four columns as well as the interval of tolerance in building regularization processes, which are 1 m, 1.5 m, 2 m, and 2.5 m consecutively. The comparison of extracting data to reference datasets obtained maximum number of errors in each sample data. The error was depicted as a percentage which is illustrated in Figure 17a. According to the graph, the maximum error was found in sample data N, where it had indefectible error regarding to line segment of building extraction. The most suitable interval tolerance value in regularization process was 1 m. Overall, the number of errors on average was 30% or 0.3, meaning the line segment of this building extraction is acceptable.



**Figure 17.** (a) Maximum error of every extracted data; (b) Shape index similarity difference.

Assessment of shape similarity resulted value of index in every sample. The result of shape similarity index is shown as Figure 17b. Related to extracted data and reference dataset there was a low difference index mostly. On average, the whole sample had 0.02 difference in shape index value. Comparing the whole data, sample S merely generated the highest difference shape index, or quantitatively was 0.3. Thus, the resulted of building extraction had a nearly similar shape to reference data in particular of building edge features.

The last test was positional accuracy using the calculation of Root Mean Square Error (RMSE). Accuracy was assessed by comparing centroid positions between extracted and reference sample objects. Centroid was used because Euclidean distance between the centers of the mass of an extracted object and the reference object could be used as another metric to measure positional accuracy other than the corner points of buildings [38]. From those buildings, centroid coordinates were calculated on both data as shown in Table 4, then RMSE was obtained from equation (9).

$$RMSE = \sqrt{\frac{\Delta X^2 + \Delta Y^2}{n}} \quad (9)$$

where  $n$  is the number of points.

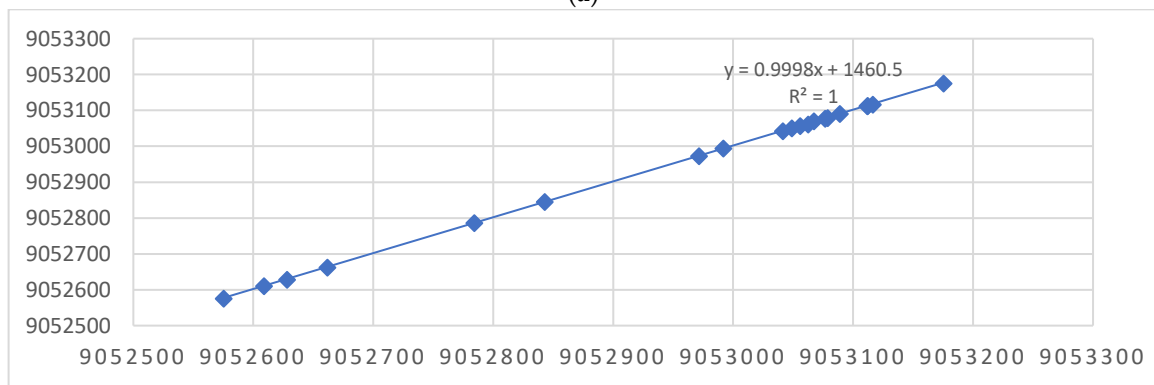
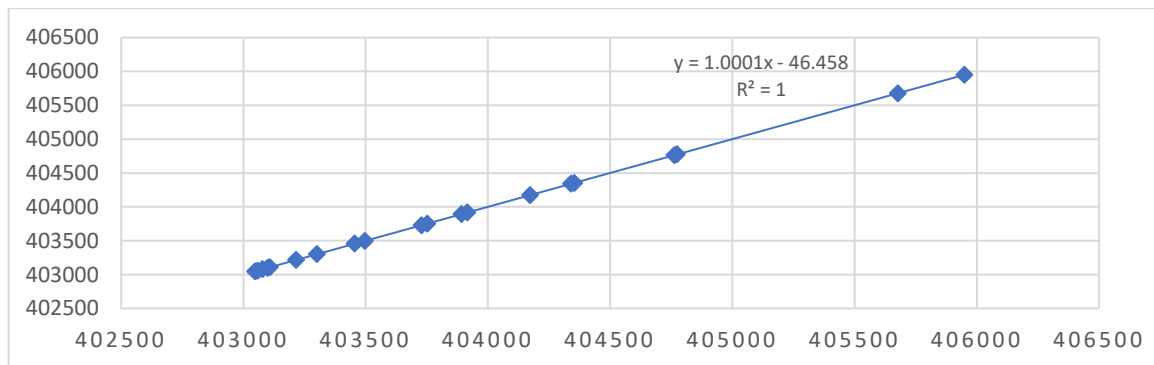
**Table 4.** Positional accuracy of each building.

Building	RBI Centroid		Extracted Centroid		$\Delta X$	$\Delta Y$
	X	Y	X	Y		
A	405948.5634	9052575.401	405948.7904	9052574.619	0.226927	-0.78278
B	403751.8255	9052608.894	403751.9463	9052609.337	0.120799	0.44350
C	404762.9299	9052628.274	404763.5870	9052627.533	0.657047	-0.74099
D	405676.4566	9052661.92	405675.4476	9052661.517	-1.00909	-0.40284

Building	RBI Centroid		Extracted Centroid		$\Delta X$	$\Delta Y$
	X	Y	X	Y		
E	404172.2934	9052784.286	404172.0479	9052785.029	-0.24550	0.74309
F	403496.5628	9052842.961	403495.4145	9052844.058	-1.14823	1.09681
G	403047.6751	9052971.569	403047.7159	9052972.250	0.040818	0.68093
H	404773.5661	9052991.895	404774.6981	9052992.894	1.131985	0.99836
I	404352.5259	9053041.472	404353.3299	9053041.220	0.804018	-0.25141
J	403099.2191	9053048.893	403098.9275	9053049.127	-0.29165	0.23374
K	403916.2759	9053055.857	403916.3749	9053055.938	0.099017	0.08112
L	403215.2586	9053062.684	403214.7439	9053060.015	-0.51466	-2.66944
M	403077.9607	9053076.599	403077.7129	9053076.649	-0.24784	0.04981
N	404339.5701	9053067.305	404340.5263	9053068.846	0.956276	1.54164
O	403728.3297	9053078.913	403728.0897	9053077.952	-0.23999	-0.96126
P	403107.3267	9053089.027	403106.5055	9053089.452	-0.82120	0.42480
Q	403892.0488	9053111.929	403892.2003	9053111.525	0.151487	-0.40353
R	403455.3598	9053116.471	403457.0229	9053115.892	1.663097	-0.57930
S	403058.2910	9053112.110	403058.0958	9053112.389	-0.19524	0.27849
T	403300.1876	9053175.343	403301.2474	9053174.690	1.059716	-0.65262

RMSE = 1,1686316

The relationship between centroid coordinates in both data can be seen in Figure 18. Due to small difference distance between coordinates from extracted and reference objects, the graph shows positive correlations. It means that based on positional accuracy using centroid accuracy assessment, this method could produce good geometric accuracy.



**Figure 18.** Correlation between coordinates from extracted and reference data; (a) Correlation between X coordinates; (b) Correlation between Y coordinates.

3.3.3. Time Processing Analysis

Our works were conducted using laptop with specification of processor Intel(R) Core (TM) i7-7500U, RAM memory 8.00 GB, and 64-bit Operating system. Thus, the result of this analysis can be a consideration for practicality evaluation using that hardware specification, and the process could be faster using higher computer specification. Time processing analysis was conducted for each step, namely ground/non-ground classification, building classification, segmentation, and building footprint extraction for an area of 2.478 km<sup>2</sup>. Based on existing topographic map, it has 3637 buildings. The result of time processing analysis is presented in Table 5.

**Table 5.** The result of speed processing analysis.

Process	Times	Information	Software
1. Ground / non-ground classification	1 minutes 51.32 seconds	1.a. Adaptive-TIN Surfaces algorithm	Terrasolid
2. Building classification	7 minutes 20.81 seconds	2.a. Training area selection	Cloud Compare
	0 minutes 24.54 seconds	2.b. Generation of training data	Cloud Compare
	7 minutes 57.62 seconds	2.c. Implementation to whole area	Cloud Compare
3. Segmentation	3 minutes 33.85 seconds	3.a. Processing of segment growing	Point Cloud Mapper
4. Building footprint extraction	16 minutes 40.22 seconds	4.a. Boundary detection	MATLAB
	0 minutes 31.95 seconds	4.b. Edge delineation	ArcGIS Pro
	0 minutes 43.13 seconds	4.c. Building footprint regularization	ArcGIS Pro
<b>TOTAL</b>	37 minutes 12.12 seconds		

According to the result above, it shows that this simple method could complete the whole process very quickly. While manual editing or digitization was still necessary, it was only needed in a few areas since more than half of entire data were statistically satisfying. The time processing analysis indicated that our method run fast, and although this proposed method was simple, it produced good results in a high effectivity manner.

4. Conclusions

This research aimed to support mapping acceleration especially for topographic map production. It focused on automatic building extraction from LiDAR data using combination of several methods, namely classification, segmentation, and building footprint extraction. Building class classification used multi-scale local dimensionality to separate points into two classes at the maximum separability plane. Building footprint extraction comprised three steps: edge points detection, building delineation, and building regularization. With 10 samples and step 0.5, classification resulted quality and miss factor of 0.597 and 0.524, respectively. The quality was improved by segmentation process to 0.604, while miss factor was getting worse to 0.561. Meanwhile, on average shape index value from extracted building had 0.02 difference and the number of errors was 30% for line segment comparison. Regarding positional accuracy using centroid accuracy assessment, this method could produce RMSE of 1.169 meters. From those statistical analysis, it can

be concluded that this method produced acceptable quality for building footprint extraction. Nevertheless, the results may be improved for the further research such as using higher density of the point cloud.

**Author Contributions:** Conceptualization, D.B.S., M.I.H., N.P., F.H., P.H.; methodology, D.B.S., M.I.H., N.P., F.H., P.H.; software, M.I.H., A.R., F.H.; validation, D.B.S., N.P., T.R.N.R.; formal analysis, D.B.S., M.I.H., N.P., F.H., P.H., T.R.N.R.; writing—original draft preparation, D.B.S., M.I.H., N.P., F.H., P.H., T.R.N.R.; writing—review and editing, A.R., R.W., supervision, A.R., R.W.; project administration, M.I.H.

**Funding:** This research was funded by State Budget (APBN) of Geospatial Information Agency in 2019, especially for research activity under Research Division.

**Acknowledgments:** The authors acknowledge Geospatial Information Agency for providing the data used in this research. We also acknowledge ITC – University of Twente, especially Dr. Sander Oude Elberink for giving LiDAR training in Indonesia and also providing the software.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Park, Y.; Guldmann, J. *Computers , Environment and Urban Systems Creating 3D city models with building footprints and LIDAR point cloud classification : A machine learning approach*, *Comput. Environ. Urban Syst.*, vol. 75, no. November 2018, pp. 76–89, 2019, doi: 10.1016/j.compenvurbsys.2019.01.004.
2. Sugihara, K.; Zhou, X.; Murase, T. *Knowledge-Based System for Automatic 3D Building Generation from Building Footprint*, in *Intelligent Interactive Multimedia: Systems and Services*, 2012, pp. 363–373.
3. Davydova, K.; Cui, S.; Reinartz, P. *Building footprint extraction from Digital Surface Models using Neural Networks*, in *Image and Signal Processing for Remote Sensing XXII*, 2016, vol. 10004, pp. 1–10, doi: 10.1117/12.2240727.
4. Huang, J.; Zhang, X.; Xin, Q.; Sun, Y.; Zhang, P. *Automatic building extraction from high-resolution aerial images and LiDAR data using gated residual refinement network*, *ISPRS J. Photogramm. Remote Sens.*, vol. 151, no. January, pp. 91–105, 2019, doi: 10.1016/j.isprsjprs.2019.02.019.
5. Antasari, I. Y. *Building Detection from Coarse Airborne Laser Scanner Data*, University of Twente, 2019.
6. Yang, H. L.; Yuan, J.; Lunga, D.; Laverdiere, M.; Rose, A.; Bhaduri, B. *Building Extraction at Scale Using Convolutional Neural Network : Mapping of the United States*, *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.*, vol. PP, pp. 1–15, 2018, doi: 10.1109/JSTARS.2018.2835377.
7. Member, J. Y. *Learning Building Extraction in Aerial Scenes with Convolutional Networks*, *IEEE*, vol. 8828, no. c, pp. 1–8, 2017, doi: 10.1109/TPAMI.2017.2750680.
8. Setiaji, D.; Harintaka. *Ekstraksi Fitur Bangunan Secara Cepat pada Foto UAV Menggunakan Metode Deep Residual Neural Network Berbasis FCN, Elipsoida*, vol. 02, no. 01, pp. 42–49, 2019.
9. Persello, C.; Stein, A. *Deep Fully Convolutional Networks for the Detection of Informal Settlements in VHR Images*, *IEEE Geosci. Remote Sens. Lett.*, vol. 14, no. 12, pp. 2325–2329, 2017.
10. Belgiu, M.; Drăguț, L. *Comparing Supervised and Unsupervised Multiresolution Segmentation Approaches for Extracting Buildings from Very High Resolution Imagery*, *ISPRS J. Photogramm. Remote Sens.*, vol. 96, pp. 67–75, 2014, doi: 10.1016/j.isprsjprs.2014.07.002.
11. Matikainen, L.; Kaartinen, H.; Hyyppä, J. *Classification Tree Based Building Detection from Laser Scanner and Aerial Image Data*, in *ISPRS Workshop on Laser Scanning and SilviLaser*, 2007, pp. 280–

287.

12. Charaniya, A. P.; Manduchi, R.; Lodha, S. K. *Supervised Parametric Classification of Aerial LiDAR Data*, in *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW'04)*, 2004.
13. Rottensteiner, F.; Trinder, J.; Clode, S.; Kubik, K. *Using the Dempster – Shafer method for the fusion of LIDAR data and multi-spectral images for building detection*, *Inf. Fusion* 6, vol. 6, pp. 283–300, 2005, doi: 10.1016/j.inffus.2004.06.004.
14. Guo, B.; Huang, X.; Zhang, F.; Shon, G. *Classification of airborne laser scanning data using JointBoost*, *ISPRS J. Photogramm. Remote Sens.*, vol. Article in, 2014.
15. Xu, S.; Vosselman, G.; Elberink, S. O. *Multiple-entity based classification of airborne laser scanning data in urban areas*, *ISPRS J. Photogramm. Remote Sens.*, vol. 88, pp. 1–15, 2014.
16. Tomljenovic, I.; Tiede, D.; Blaschke, T. *A building extraction approach for Airborne Laser Scanner data utilizing the Object Based Image Analysis paradigm*, *Int. J. Appl. Earth Obs. Geoinf.*, vol. 52, pp. 137–148, 2016, doi: 10.1016/j.jag.2016.06.007.
17. Du, S.; Zhang, Y.; Zou, Z.; Xu, S.; He, X.; Chen, S. *Automatic building extraction from LiDAR data fusion of point and grid-based features*, *ISPRS J. Photogramm. Remote Sens.*, vol. 130, pp. 294–307, 2017, doi: 10.1016/j.isprsjprs.2017.06.005.
18. Widyaningrum, E.; Gorte, B.; Lindenbergh, R. *Automatic Building Outline Extraction from ALS Point Clouds by Ordered Points Aided Hough Transform*, *Remote Sens.*, 2019, doi: 10.3390/rs11141727.
19. Liu, C.; Shi, B.; Yang, X.; Li, N.; Wu, H. *Automatic Buildings Extraction From LiDAR Data in Urban Area by Neural Oscillator Network of Visual Cortex*, *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.*, pp. 1–12, 2012.
20. Huang, H.; Brenner, C.; Sester, M. *A generative statistical approach to automatic 3D building roof reconstruction from laser scanning data*, *ISPRS J. Photogramm. Remote Sens.*, vol. 79, pp. 29–43, 2013, doi: 10.1016/j.isprsjprs.2013.02.004.
21. Awrangjeb, M.; Fraser, C. *Rule-based Segmentation of LiDAR Point Cloud for Automatic Extraction of Building Roof Planes*, *ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci.*, vol. II-3, no. November, pp. 1–6, 2013, doi: 10.5194/isprsannals-II-3-W3-1-2013.
22. Awrangjeb, M.; Ravanbakhsh, M.; Fraser, C. S. *Automatic Detection of Residential Buildings using LIDAR Data and Multispectral Imagery*, *ISPRS J. Photogramm. Remote Sens.*, vol. 65, no. 5, pp. 457–467, 2010, doi: 10.1016/j.isprsjprs.2010.06.001.
23. Awrangjeb, M.; Zhang, C.; Fraser, C. S. *Automatic Extraction of Building Roofs using LIDAR Data and Multispectral Imagery*, *ISPRS J. Photogramm. Remote Sens.*, vol. 83, pp. 1–18, 2013, doi: 10.1016/j.isprsjprs.2013.05.006.
24. Cheng, L. *et al. Integration of LiDAR Data and Optical Multi-view Images for 3D Reconstruction of Building Roofs*, *Opt. Lasers Eng.*, vol. 51, no. 4, pp. 493–502, 2013, doi: 10.1016/j.optlaseng.2012.10.010.
25. Wang, R. *3D building modeling using images and LiDAR: a review*, *Int. J. Image Data Fusion*, vol. 9832, 2013, doi: 10.1080/19479832.2013.811124.
26. Kada, M.; McKinley, L.; Systems, V. C. *3D Building Reconstruction from LiDAR Based on a Cell Decomposition Approach*, in *Stilla U, Rottensteiner F, Paparoditis N (Eds) CMRT09. IAPRS*, 2009, vol. XXXVIII, pp. 3–4.
27. Li, M.; Rottensteiner, F.; Heipke, C. *Modelling of buildings from aerial LiDAR point clouds using TINs*

- and label maps, *ISPRS J. Photogramm. Remote Sens.*, vol. 154, no. May, pp. 127–138, 2019, doi: 10.1016/j.isprsjprs.2019.06.003.
28. Sun, S.; Member, S.; Salvaggio, C. *Aerial 3D Building Detection and Modeling From Airborne LiDAR Point Clouds*, *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.*, vol. 6, no. 3, pp. 1440–1449, 2013, doi: 10.1109/JSTARS.2013.2251457.
  29. Susetyo, D. B.; Hidayat, F.; Hariyono, M. I. *Automatic Building Model Extraction using LiDAR Data*, in *The 39th Asian Conference on Remote Sensing "Remote Sensing Enabling Prosperity,"* 2018, pp. 1686–1695.
  30. Wen, X.; Xie, H.; Liu, H.; Yan, L. *Accurate Reconstruction of the LoD3 Building Model by Integrating Multi-Source Point Clouds and Oblique Remote Sensing Imagery*, *Int. J. Geo-Information*, vol. 8, no. 135, 2019, doi: 10.3390/ijgi8030135.
  31. Ramiya, A. M.; Nidamanuri, R. R.; Krishnan, R. *Segmentation based building detection approach from LiDAR point cloud*, *Egypt. J. Remote Sens. Sp. Sci.*, vol. 20, no. 1, pp. 71–77, 2017, doi: 10.1016/j.ejrs.2016.04.001.
  32. Axelsson, P. *DEM Generation from Laser Scanner Data Using Adaptive TIN Models*, *Int. Arch. Photogramm. Remote Sens.*, vol. 23, no. B4, pp. 110–117, 2000, doi: 10.1016/j.isprsjprs.2005.10.005.
  33. Rizaldy, A.; Mayasari, R. *Acceleration of Topographic Map Production Using Semi-Automatic Dtm From Dsm Radar Data*, *ISPRS - Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.*, vol. XLI-B7, no. July, pp. 47–54, 2016, doi: 10.5194/isprsarchives-XLI-B7-47-2016.
  34. Brodu, N.; Lague, D. *3D terrestrial lidar data classification of complex natural scenes using a multi-scale dimensionality criterion : Applications in geomorphology*, *ISPRS J. Photogramm. Remote Sens.*, vol. 68, pp. 121–134, 2012, doi: 10.1016/j.isprsjprs.2012.01.006.
  35. Vosselman, G. *Point Cloud Segmentation for Urban Scene Classification*, in *ISPRS2013-SSG*, 2013, no. October 2013, doi: 10.5194/isprsarchives-XL-7-W2-257-2013.
  36. Rutzinger, M.; Rottensteiner, F.; Pfeifer, N. *A Comparison of Evaluation Techniques for Building Extraction From Airborne Laser Scanning*, *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.*, vol. 2, no. March 2009, 2009, doi: 10.1109/JSTARS.2009.2012488.
  37. Gribov, A. *Searching for a Compressed Polyline with a Minimum Number of Vertices*, *arXiv*, 2015.
  38. Zeng, C.; Wang, J.; Lehrbass, B. *An Evaluation System for Building Footprint Extraction From Remotely Sensed Data*, *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.*, vol. 6, no. 3, pp. 1640–1652, 2013.
  39. Kweon, H.; Kim, M.; Lee, J.; Seo, J. Il; Rhee, H. *Comparison of Horizontal Accuracy , Shape Similarity and Cost of Three Different Road Mapping Techniques*, *Forest*, pp. 1–13, 2019, doi: 10.3390/f10050452.
  40. Veltkamp, R. C. *Shape Matching : Similarity Measures and Algorithms*, in *Proceedings International Conference on Shape Modeling and Applications*, 2011.
  41. Veltkamp, R. C.; Hagedoorn, M. *Shape Similarity Measures, Properties and Constructions*, in *Advances in Visual Information Systems*, 2000, pp. 467–476.
  42. Ramirez, J. R.; Ali, T. *Progress in Metrics Development to Measure Positional Accuracy of Spatial Data*, in *Proceedings of the 21th International Cartographic Conference (ICC)*, 2003, no. August.