

Article

Developing an Algorithm for Buildings Extraction and Determining Changes from Airborne LiDAR, and Comparing with R-CNN Method from Drone Images

Saied Pirasteh ^{1,2,*}, Pejman Rashidi ³, Heidar Rastiveis ³, Shengzhi Huang ¹, Qing Zhu ¹, Guoxiang Liu ¹, Yun Li ¹, Jonathan Li ² and Erfan Seydipour ³

¹ Department of Surveying and Geoinformatics, Faculty of Geosciences and Environmental Engineering, Southwest Jiaotong University (SWJTU), the Western Park of the Hi-Tech Industrial Development Zone, Chengdu, Sichuan 611756, China; huangsz@my.swjtu.edu.cn (S.H.); zhuq66@263.net (Q.Z.); rsgxliu@swjtu.edu.cn (G.L.); liyun20151202@my.swjtu.edu.cn (Y.L.)

² Mobile Sensing and Data Science Lab, University of Waterloo, 200 University Ave., Waterloo, ON N2L 3G1, Canada; junli@uwaterloo.ca

³ School of Surveying and Geospatial Engineering, College of Engineering, University of Tehran, Tehran 14174-66191, Iran; pejman.rashidi@ut.ac.ir (P.R.); hrasti@ut.ac.ir (H.R.); erfan.seidipoor@ut.ac.ir (E.S.)

* Correspondence: spirasteh71@gmail.com or sapirasteh@swjtu.edu.cn

Received: 30 April 2019; Accepted: 23 May 2019; Published: 29 May 2019



Abstract: The world has experienced urban changes rapidly, and this phenomenon encourages authors to contribute to the United Nations sustainable development goals (SDGs) 2030 and geospatial information. This study presents a proposed algorithm of change detection and extracting the borders of buildings. This proposed algorithm provides a set of instructions to describe the method of solving the problem of how extracting the boundary of buildings from the light detection and ranging (LiDAR) input data incorporating with the firefly and ant colony algorithms. The method has used two different epochs to compare buildings and to identify the type of changes in selected buildings. These changes are based on the newly built or demolished buildings. We also used drone images and mask the region-based convolutional neural network (R-CNN) method to compare the results of roof extraction of buildings vs. the proposed algorithm. This study shows that the proposed algorithm identifies the changes of all buildings with higher accuracy of extracting border of buildings than the existing methods, successfully. This study also determines that the amount of root mean square error (RMSE) is 2.40 m^2 when we use LiDAR. This proposed algorithm contributes to identifying rapidly changed buildings, and it is helpful for global geospatial information of urban management that can add best practice and solution toward the UN SDGs connectivity dilemma of urban settlement, resilience, and sustainability.

Keywords: LiDAR; change detection; border extraction; firefly algorithm; ant colony algorithm; drone images; R-CNN

1. Introduction

The future trends in geospatial information management have a great impact on urban development and changes. The role of geospatial information such as airborne light detection and ranging (LiDAR) point clouds in measuring and monitoring changes can support the 2030 agenda for sustainable development. Therefore, this study attempted to implement the use of point clouds in 3D change detection (CD) of buildings that ultimately benefits research toward a contribution to the 2030 agenda for sustainable development.

Building change detection (CD) is one of the most important processes in monitoring and managing urban development. It has a wide range of applications varies from damage assessment to urban management [1–3]. Furthermore, efficient change detection is based on minimal data sources that are essential for updating changes to support time-critical applications such as disaster response and recovery [1,4].

Over the last few decades, many studies have used 2-dimensional (2D) space-borne and airborne remote-sensing images on large-scale problems such as urban sprawl, forest monitoring, and natural hazards [5–7]. Although the aerial imagery has been employed for map compilation in updating process, it has been seen that the characters cause some unavoidable problems [2,8,9]. For instance, casting shadows to dominate in the scenes acquired with many buildings over dense urban areas, the spectral information of buildings in aerial images is diverse and ill-defined. Therefore, perspective projection causes the leaning of buildings and skyscrapers, which requires height information to correct it [2,10,11]. It is worth noting that the 3-dimensional (3D) geometric information reflects the physical geometry of the objects that has a great potential to improve the performance and overcome some of the restrictions of the traditional 2D image-based CD [12].

Three-dimensional change detection has vividly more advantages in comparison to the limitations of 2D change detection for example; the 3D geometric information is free of illumination variations and perspective distortions. Despite these advantages, the major barrier to 3D change detection applications was the cost and accessibility of 3D data. Airborne LiDAR flights were usually expensive and high accuracy photogrammetric stereo measurements from airborne or space-borne images have still required manual involvement [1–3]. However, these days there is much better access to accurate 3D data. In addition, Radovic et al. (2017) studied detailed procedures and parameters for the training of convolutional neural networks (CNNs) for efficient and automated object recognition on the aerial photos. They found that the accuracy and reliability of CNNs depend on the network's training and how to select operational parameters [13]. Also, Uijlings studied recent advances in object detection [14]. It is driven by the success of region proposal methods and the region-based convolutional neural networks (R-CNNs) [14]. These region-based CNNs are computationally expensive as compared to the approach that was originally developed by Girshick [15]. They reduced the cost drastically to sharing convolutions across proposals given by Girshick and He [16,17].

There are some limitations and challenges in 2D image-based CD and most of them are perspective distortion, higher spectral variability, and lack of volumetric information [12]. Some methods used subtraction between two digital surface models (DSMs) in order to find change detection. The first study in building change detection was based on DSM comparisons [13,14]. However, there are different methods for CD using airborne LiDAR data and drone images (Table 1). There are a lot of studies that are based on the DSMs to detect changes and they are highly dependent on the quality of the DSM [17–19]. It is worth noting that analysis of solely a dDSM (DSM (t_1)–DSM (t_2)) can lead to ambiguities, for example, attachments or modifications of buildings are not related to the affected buildings [20]. The motivation of the authors to apply this proposed method is because they are sensitive to misregistration and may produce some false positives for matched DSMs. Moreover, some studies are based on computing the Euclidian differences between two 3D surfaces. They are time consuming for corresponding searches and also need complicated implementation [12].

Nevertheless, there are many weaknesses in some methods for building change detection, especially in methods based on DSMs and Euclidian differences. Therefore, this study presents the development of an algorithm of extraction of the boundary of buildings and how nDSM is generated by filtering techniques to detect changes of buildings. Furthermore, we used R-CNN for drone images classification of object detection for a selected region in China. This is because R-CNN has been attracting increasing attention for efficient, yet accurate, and visual recognition. In addition, R-CNN applies for both objection detection tasks and region proposal generation as well. Perhaps, with such a design we could detect objects much faster than other methods when we use drone images [18]. We also compared mask R-CNN with the proposed algorithm.

Table 1. List of proposed change detection using airborne light detection and ranging (LiDAR) data.

Authors	Year	Change Detection [CD] Approach
Vu et al.	2004 [11]	The detected results were visually evaluated by using ortho-image produced by LiDAR surveying flights. In this method, to extract the possible new construction and demolition, a histogram thresholding was applied.
Teo and Shih	2013 [21]	Proposed a method based on differential DSM (dDSM). The intersection of normalized DSM (nDSM) and dDSM was used to detect and locate the candidate changed regions. In this method, there are some errors that occurred in small and vegetation areas.
Xu et al.	2015 [22]	They used a rule-based classifier combined with the point to plane distance between two epochs in order to find “changed”, “unchanged”, and “unknown”. Eventually, they showed that the quality of the classification results will influence the quality of the CD.
Xu et al.	2015 [23]	After registering and filtering, an octree is generated to store and index the irregularly LiDAR data. Then, comparing the LiDAR point clouds to find changes in building and trees.
Tran et al.	2018 [24]	Used machine learning to find CD. This method used for finding changes in building, tree, and ground. However, there is not any method for extracting border of buildings.
Ren et al.	2015 [18]	Region proposal network (RPN) was introduced. It shares full image convolutional features to detect objects.

2. Study Area and Data Used

Utah State in the USA is taken as the study area. Because of the extreme topography and different objects (such as road, car, vegetation, and buildings with different sizes) the experimental region (Figure 1) is generally challenging. The airborne LiDAR data in this study was obtained from the publicly available National Science Foundation (NSF) Open Topography website [25,26]. This material is based on services provided to the Plate Boundary Observatory (PBO) by Airborne Laser Swath Mapping from National Center for Airborne Laser Mapping (NCALM). PBO is operated by UNAVCO for Earth Scope and is supported by the National Science Foundation. Two test samples were downloaded from the NSF Open Topography website. They were obtained on 8 April 2008, and 18 October 2013, and have average point densities of 6.74 pts/m² and 11.93 pts/m², respectively. It is worth noting that the second set of data have RGB information. This study area is large enough (it has 110 buildings), and it contains different features that are common in an urban area.

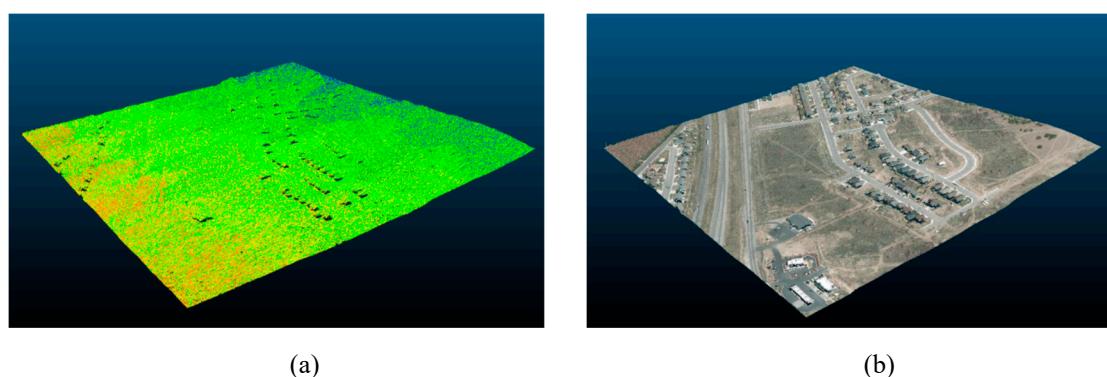


Figure 1. Experimental area. (a) LiDAR data for the year 2008; (b) LiDAR data for the year 2013.

In addition to the aforementioned, we used three clips from a Document Object Model (DOM) image (Figure 2) that are generated by drone images from Bantian Residential District, located in Longgan Shenzhen, Guangdong, China. These drone images were taken from the Feisi IQ 180 camera with the 0.067m per pixel Ground Sample Distance (GSD).

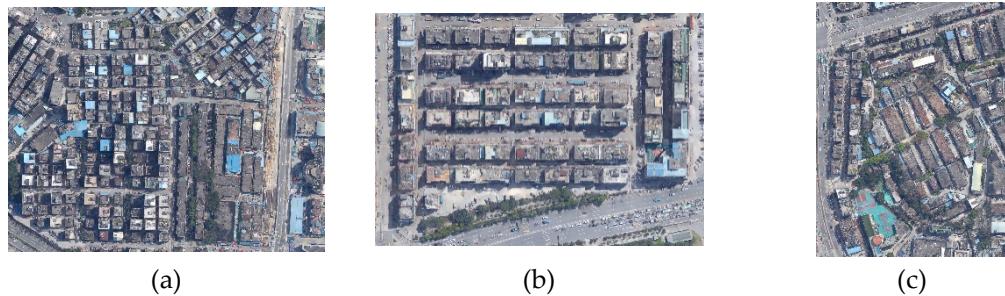


Figure 2. Experimental area of China. (a) region 1, (b) region 2, and (c) region 3 from drone images.

3. Methodology

In this study, the airborne LiDAR data was chosen because there are some advantages such as data acquisition of large areas in a relatively short time or extensive independence of weather and lighting conditions. Although most studies have analyzed dDSM for change detection that it would lead to some ambiguities. In this study, the authors extracted buildings separately from each LiDAR dataset and have compared them with another dataset. Also, most studies such as [21–25], did not have any suggestion for the border of building extraction after the change detection; however, this study provides a solution by proposing a new algorithm. The proposed method consists of five steps as shown in Figure 3. As can be seen in this figure, the raw LiDAR point clouds are preprocessed to eliminate outliers accurately register two airborne LiDAR data, and distinguished ground and non-ground points. Then, a normalized digital surface model (nDSM) was generated to extract building points after applying the height threshold accurately. Finally, by comparing the extracted buildings, the type of changes was defined, and the borders of buildings were determined.

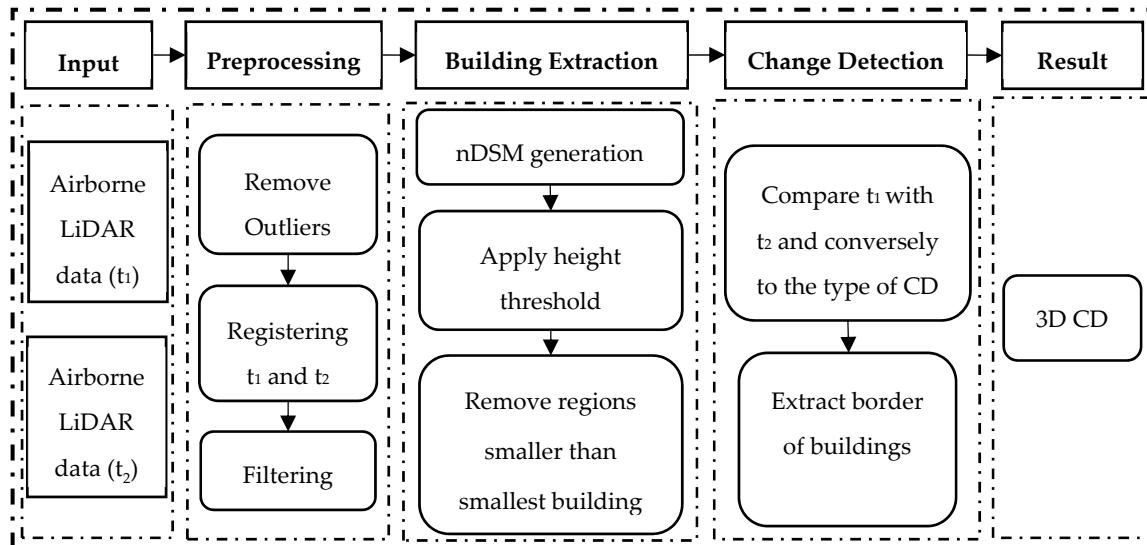


Figure 3. Flowchart of the proposed algorithm.

3.1. Proposed Algorithm

3.1.1. Preprocessing

The proposed algorithm describes the instructions of the method of how extracting the boundary of buildings from the LiDAR input data incorporating with the firefly and ant colony algorithms. The first step of instruction of the method is the preprocessing. The preprocessing is implemented to remove the outliers from the data. The outliers of LiDAR data are points with abnormal elevation values, either higher or lower than the surrounding points. The outliers with high elevation values

which usually include random errors and result from birds or airplanes were eliminated during the filtering process because they could be assumed to be non-ground objects. While low outliers are below the surface and may be result from several times reflecting of the laser returns. These outliers may seriously affect the filtering results. Therefore, they should be removed from the data in the preprocessing step [27–30].

One of the most significant approaches for removing low outliers is to compute the vertical difference between the height of each point and the average height of its neighboring points [31]. Therefore, a local window is proposed to consider the algorithm instruction and it can be seen in Equation (1). The height of points $H_i^p \in H^p, i \in 1 \dots n$ are sorted by a function s .

$$H_{s(1)}^p \leq \dots \leq H_{s(n)}^p \quad (1)$$

After sorting the height of points in a local window, all of them are compared with the average height. Then, the points that are lower than the average with regards to a predefined threshold (10 m) and are potentially low outliers should be removed from the data.

The next step in the proposed algorithm instruction and the preprocessing is co-registration. The registration procedure can be done automatically or manually [21]. In this study, the initial registration is done by Cloud Compare software version 2.6.3, but because of some distinctions, manual registration is applied as well. An object with height discontinuities is not suitable for selecting a registration point; therefore, in order to process registration manually, the measurement of points on the road surface has been considered into the account [21].

The last step from the algorithm instruction in the preprocessing is LiDAR point clouds filtering. In the digital elevation model (DEM) generation from the LiDAR data, the first step is to determine and distinguish the ground and non-ground points, a process referred to as filtering [32]. In this regard, a method based on the slope and progressive window thresholding (SPWT) has been used [29]. In this method, the non-ground points are eliminated through three main steps: (a) small window thresholding, (b) slope thresholding, and (c) large window thresholding [29].

The SPWT algorithm is designed to work on a grid form of the LiDAR data. Therefore, the first step is re-sampling of the LiDAR point clouds into a regular grid data. Then, in a small window, the elevation difference between the candidate pixel and the minimum elevation of the local window is calculated to detect the non-ground pixels. Afterward by considering a predefined threshold value of the slope between each pixel and its neighbor pixels the non-ground pixels which have not been previously recognized are detected. The last step is the same as the aforementioned step, but it is in a larger window size. Actually, the elevation difference between the candidate pixel and the minimum elevation of the local window is calculated. If the difference value exceeds from a predefined threshold, the candidate pixel is labeled as non-ground point [29].

The window sizes are specified from the smallest to the largest objects in the area. In addition, according to the physical characteristics of the ground surface and the size of objects, the best height threshold is selected manually. Furthermore, the slope threshold should be assigned based on the topographical condition of the area. The SPWT method selects the pixels in order from the first to the last scan line, and after finding the ground seed the algorithm iterates repeatedly for all points through the following steps to the label points as ground or non-ground.

3.1.2. Building Extraction

In order to extract buildings, various methods have been proposed to distinguish buildings from trees and noise [21]. One of these methods is the removal of all regions with an area smaller than the smallest building [21,33,34]. In other words, patches that cover small areas tend to be classified as commission errors and removed by an area threshold [21]. As a result, the boundary of the buildings in regions are enhanced and defined.

After the filtration of the LiDAR point clouds in the algorithm instruction of the preprocessing step, the first step in building extraction is the DEM generation. For this purpose, the authors used the ground points that are remained as the post-filtering process. Therefore, it is necessary to resample gaps through an interpolation technique after removal of the non-ground points. In this study, the nearest neighbour technique has been implemented. This technique considers the elevation of the nearest point in a specified distance to the output pixel. If no point is observed in the specified distance, the pixel would be labeled as no data. Therefore, to avoid too many or no points in each grid cell, the process allows determining the size from the average point spacing of the point clouds [35].

In the next step, when the DEM is built the process allows transmitting all objects to a smooth surface. This transmission allows generating the nDSM by implementing Equation (2) in the proposed algorithm. The nDSM is useful because objects with a high altitude such as buildings and trees can be recognized through an appropriate threshold; also, with a decent filtering method, the topography does not have an adverse effect on it. Indeed, the nDSM was generated to describe the above-ground height information [21,36].

$$nDSM = DSM - DEM \quad (2)$$

3.2. Change Detection (CD)

Change detection in an urban area is mostly classified as “changed” or “unchanged.” However, because of using LiDAR point clouds there is an opportunity for detecting changes in height. In other words, not only 2D change detection such as newly built and/or demolished building can be detected, but also detecting changes in height (e.g., converting a two-story building to three floors) is possible.

In this study, after recognition of the building regions from the LiDAR data in two different time epochs in the same area, the buildings from each epoch of time are compared with other buildings from another dataset. In fact, the authors have seen that the “changed-buildings” are buildings that have a different shape in two periods—the “newly built” divided into two categories: (1) newly built from existing building to building, (2) newly built from non-building to building [21]. The newly built from existing building to building represents the fact that an old building was destroyed in the past and replaced with a new building in the latter period. The newly built from non-building to building means a new building is established that has not existed before.

It is worth noting that all characteristics of changes are defined by comparing the attributes of the regions. If the regions in the past period do not have corresponding regions in the latter time period, these regions can be defined as demolished buildings. Conversely, regions in the latter period that do not have corresponding regions in the past period are defined as newly built. It has been clearly seen that the newly built buildings are vividly shown by large changes in an area. Nevertheless, this study was compared the regions as buildings separation to find the type of changes. Finally, the borders of buildings are detected and the changed-buildings are determined.

3.3. Extraction of Building Boundary: Incorporating with Firefly and Ant Colony Algorithms

After finding patches as building regions, a rectangle or some rectangles are fitted to each building. For this purpose, buildings are assumed that have a vertical corner. In this study, borders of buildings are extracted incorporating with the firefly and ant colony algorithms. In addition, the proposed algorithm has been compared with the borders of buildings that were extracted manually by using ArcGIS software version 10.2. The following explains incorporation of the firefly and ant colony algorithm concepts within the proposed algorithm.

3.3.1. Firefly Algorithm

The firefly algorithm is based on the flashing patterns and behavior of fireflies. This algorithm is based on the movement and it continues to fit the best shape on the data. The movement of a firefly i is attracted to another more attractive or brighter firefly j is determined by Equation (3) [37].

$$x_i^{t+1} = x_i^t + \beta_0 e^{-\gamma r_{ij}^2} (x_j^t - x_i^t) + \alpha_t \varepsilon_i^t \quad (3)$$

where, the second term is because of the attraction. The third term is randomization with α_t being the randomization parameter, and the last term (ε_i^t) is a vector of random numbers drawn from a Gaussian distribution or uniform distribution at time t [31]. In Equation (3), the variation of attractiveness β is defined with the distance r , and β_0 is the attractiveness at $r = 0$. Also, if $\gamma = 0$, it reduces to a variant of particle swarm optimization. The firefly algorithm needs a target function to fit rectangle on building region. Here, the target function is based on Equation (4).

$$Q = (A - B) + (B - A) - (A \cap B) \quad (4)$$

where, A is the total pixels in each rectangle and B are the total pixels that belong to each region of the building. Also, the target function (Q) should reach the lowest possible value. In order to fit the best rectangle on each region, the firefly algorithm is incorporated in the proposed method to determine five parameters including the position of one of the building corners (X, Y), the size of the sides (large and small sides), and rectangular. These parameters are considered as input to the algorithm. The steps to find the rectangles of the building continue to the number of pixels remaining from the building region and to be less than a certain amount.

3.3.2. Ant Colony Algorithm

After fitting rectangles on each region of the building, the final border of the building should be determined. In this study, the authors incorporated the ant colony algorithm and also considered a solution based on the traveling salesman problem (TSP). The objective of solving the traveling salesman problem is finding the path that the seller can handle all cities in the shortest possible path [38]. Therefore, in this problem, the shortest path that passes the sides of the rectangles should be determined as the border of the building. In Figure 4, the shortest path that passes the sides of three rectangles is shown by green color.

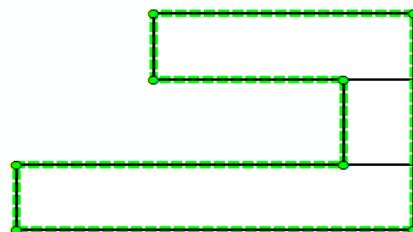


Figure 4. The shortest path that passes the sides of three rectangles by green color.

One of the best algorithms for this problem is the ant colony algorithm. This algorithm is based on ants' behavior in nature for finding food with the shortest path to their nest. They do this work by exploiting pheromone information, without using visual cues [39]. The amount of pheromone that ants put from themselves on the ground depends on the better path. The best path in the ant colony algorithm is usually the shortest path [40]. The ant colony algorithm is based on Equation (5).

$$P_k(i, j) = \begin{cases} \frac{[\tau(i, j)]^\alpha [\eta(i, j)]^\beta}{\sum_{(i, j) \notin \text{Tabu}} [\tau(i, j)]^\alpha [\eta(i, j)]^\beta} & \text{if } (i, j) \notin \text{Tabu} \\ 0 & \text{if } (i, j) \in \text{Tabu} \end{cases} \quad (5)$$

where, τ is the pheromone and for each ant K , $P_k(i, j)$ is the probability of moving from state i to another state j . Also, α, β are used to establish the relative influence of η versus τ . In addition, Tabu is the list of forbidden paths, and the ants should not move on these paths.

In this study, the ant colony algorithm is incorporated in the proposed method to finding the shortest path based on points behind in direction of the rectangle. To resolve the issue, the cost function is the smallest polygon or the shortest path. The distance between points after applying the above constraint can be achieved from Equation (6).

$$D = \sum d_{ij} + \sum d_{ij} \times \Delta a \quad (6)$$

$$\Delta a = \min(a_l - a_{ij}, a_s - a_{ij}) \quad (7)$$

In Equation (6), D is the distance between two points and should be the minimum amount. Also, d_{ij} is Euclidean distance, and as it can be seen from Equation (7), Δa must be the least value. In Equation (7), a_l is the slope of the large side in the larger rectangle which is located on the building, and a_s is the slope of the small side in the same rectangle. Ultimately, after implementing this algorithm, buildings that are collections of several rectangles, appear as an integrated polygon.

3.4. Mask Region-Based Convolutional Neural Network (R-CNN)

He [41] studied the Mask R-CNN by adding a branch for predicting segmentation masks on each region of interest (RoI), so that the result is able to achieve the pixel-to-pixel manner between network inputs and outputs. The mask branch is based on the bounding box detected by faster R-CNN, which is a small FCN in order to predict a segmentation mask. He also introduced the RoI align to fix the misalignment, which could improve mask accuracy. By combining the faster R-CNN the result performed good on the segmentation and classification for building roof detection [18].

The images were taken by a drone for particular buildings and the R-CNN method was applied to extract the bounding box and to mask the rectangle of the buildings. Figure 5a shows the bounding box mask building area and it is more precisely than the actual extracted rectangle bounding box of the building. It means that we get polygon from the mask area when the mask R-CNN is used. A reference of training dataset was used for validating the outcomes from the R-CNN method on the drone images. The mask R-CNN result was also validated from the ground truth data. Then we compared the bounding box results extracted from the LiDAR point cloud and the drone images. Figure 5b depicts the bounding box extracted from the LiDAR point cloud by using the proposed algorithm.



Figure 5. (a) Sharpness of the polygon, (b) building roof bounding box with mask.

4. Results and Discussion

In this study, a notebook with Intel Core i7 quad-core processor with 8 GB RAM is used. The proposed method is implemented in MATLAB R2015 on the Windows 10 operating system.

The basis of this research is on building extraction from each LiDAR dataset and comparing them in identifying the changes. In the preprocessing step from the proposed algorithm, a local window with a size of 10 m was considered. All points in this window were compared to the average height for

the given window and those points that were lower than the average with regards to a predefined threshold (10 m) were low outliers and removed from the data.

It is worth noting that two datasets were almost registered and Cloud Compare software was used for having better registration. Furthermore, because of some distinctions, 6 points from the corner of some buildings were used for manual processing and finally to register the points. The SPWT method was applied for LiDAR point cloud filtering after the LiDAR datasets (0.5 m as pixel size) registration process. This study indicated the values in the SPWT algorithm for filtering the LiDAR data. Table 2 summarizes the applied parameters in testing the algorithm.

Table 2. Parameter values in the slope and progressive window thresholding (SPWT) algorithm for filtering the LiDAR dataset.

Size of Small Window	Height Difference Threshold in the Small Window	The Slope Threshold	Size of Large Window	Height Difference Threshold in the Large Window
3×3	2.5 m	45°	20×20	6 m

This study resulted in determining and separating the ground and non-ground points and to generate a DEM. The nearest neighbor interpolation algorithm has also filled the gaps in creating a DEM and nDSM. This study showed that by subtracting the DSM and DEM (Figure 6), all objects can be transmitting on a smooth surface.

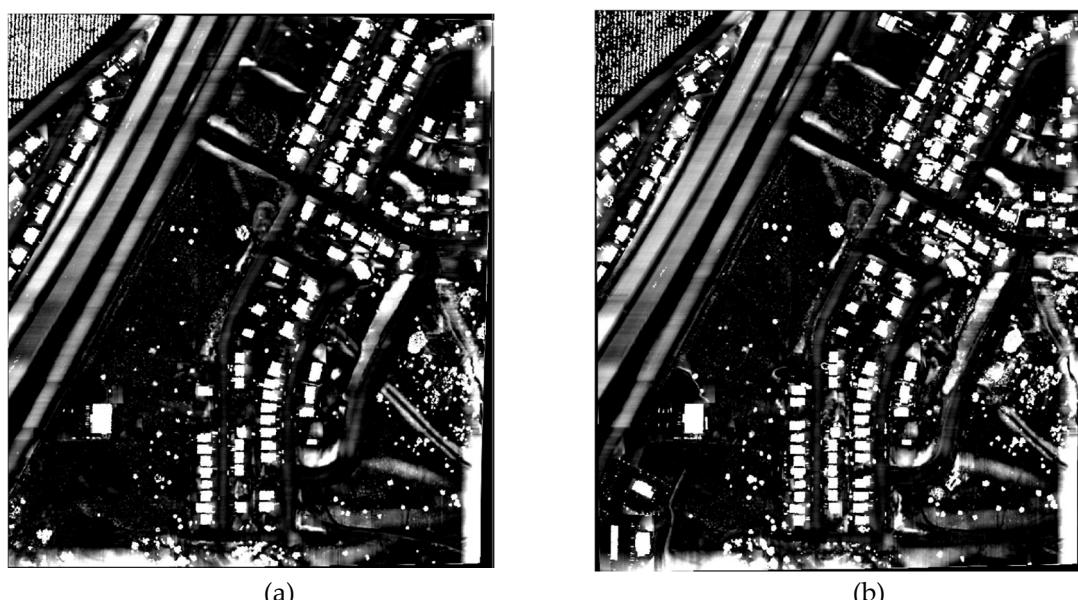


Figure 6. Output of the normalized digital surface model (nDSM). (a) nDSM for the first data; (b) nDSM for the second data.

Figure 7 depicts that a height threshold of 3.5 meter is considered to separate high altitude objects such as buildings and trees from other objects. This height threshold is considered according to the minimum height for a one-story building. Therefore, by using this threshold, no building will be removed from the data.

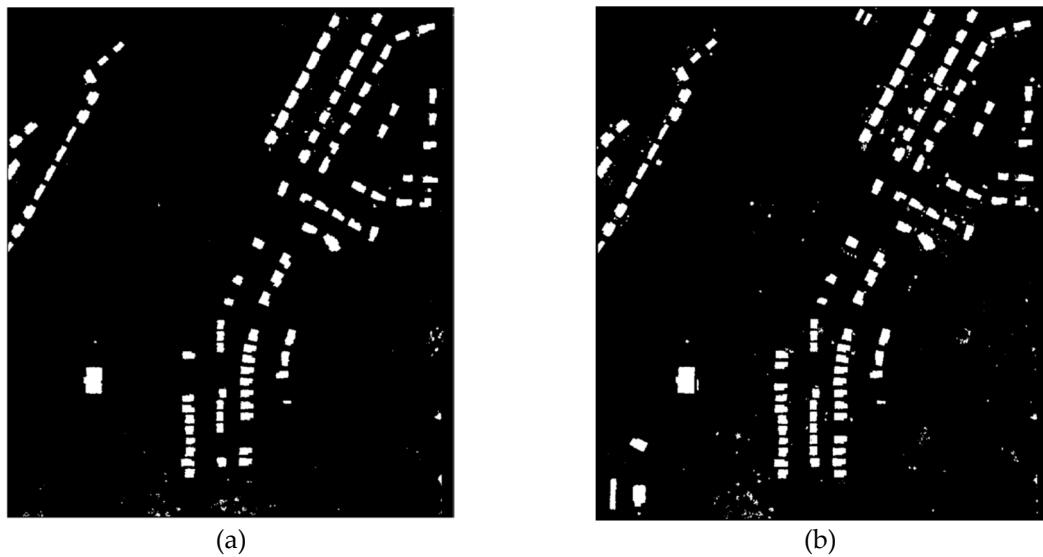


Figure 7. Output of the applying threshold on nDSM. (a) first data; (b) second data.

As it can be seen from Figures 6 and 7, most of the vegetation in the study area is shrubs and they were removed after applying the height threshold. Also, trees in this area are scattered and they are not beside the buildings. Therefore, buildings are extracted after removing regions that are smaller than the smallest building. In this study, the smallest building is 75 m^2 . The extracted buildings are shown in Figure 8 and all buildings which are 110 in the study area have been successfully extracted from the data.

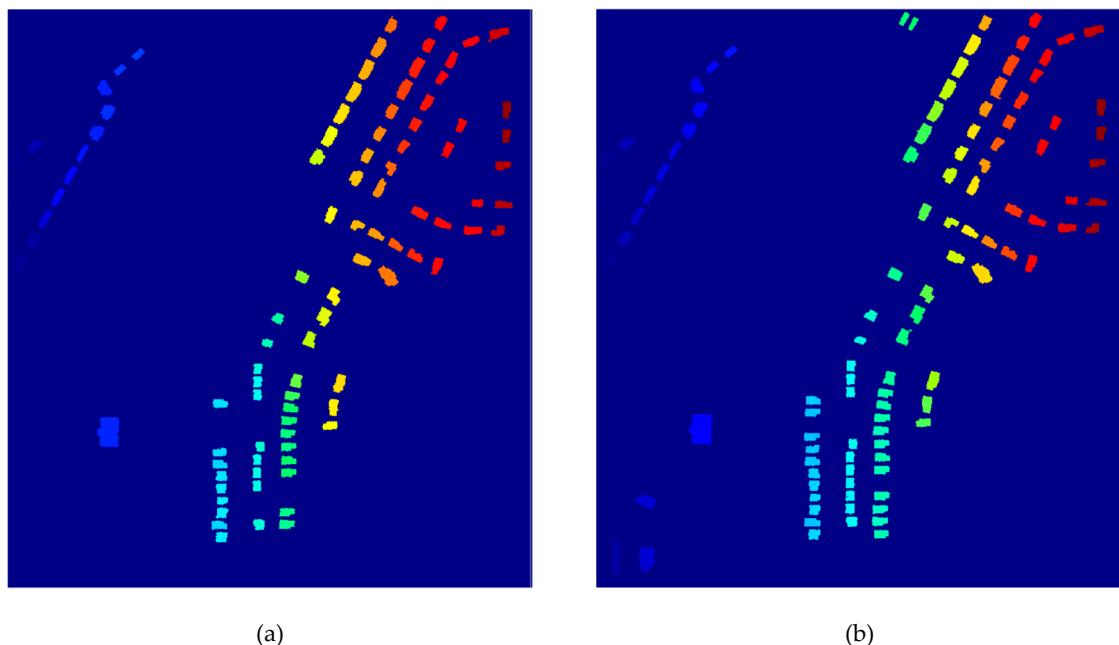


Figure 8. Buildings extracted from LiDAR data. (a) first data and (b) second data.

In order to detect the changes and finding the changed buildings, there are two different categories: (1) newly built, (2) demolished building. In this study, there are 110 buildings, and 11 buildings recognized as newly built buildings and the study shows that there is not any demolished building.

This study showed that when using 15 pixels in the proposed algorithm as the threshold to stop, the firefly algorithm had found a new rectangle. Therefore the number of repetitions threshold is 65, the population firefly threshold is 25, and the light absorption coefficient is 1. These thresholds are

considered by test and error. Also, in Figure 9, two building regions are shown whose shapes are simulated by two rectangles.

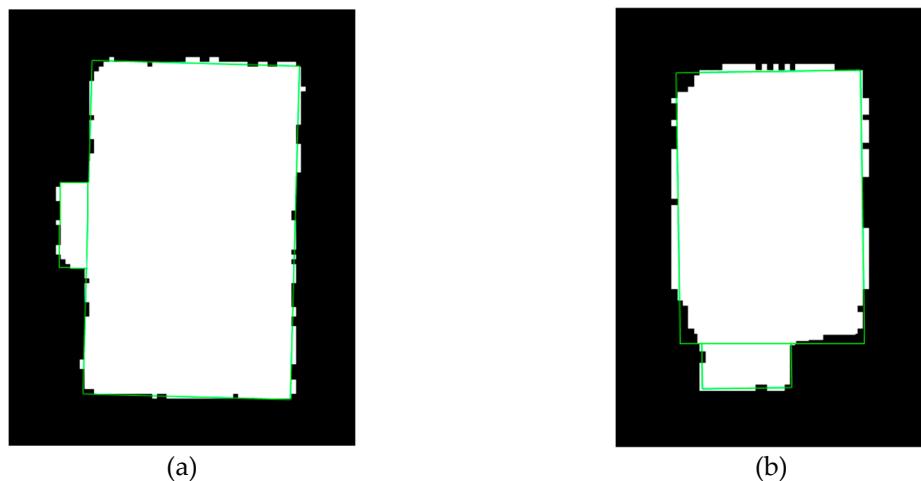


Figure 9. Fitting rectangles on building regions by green color using the firefly algorithm. (a) First simulated rectangle and (b) second simulated rectangle.

This study has simulated building regions by using some rectangles to determine the border of buildings. The proposed method also showed that using 150 ant thresholds with the coefficient of 0.05 and 30 thresholds to stop the ant algorithm could possibly enhance the building boundary better than the existing methods. These thresholds are considered by test and error as well. Figure 10 illustrates the final border of the building regions that has been extracted from Figure 9.

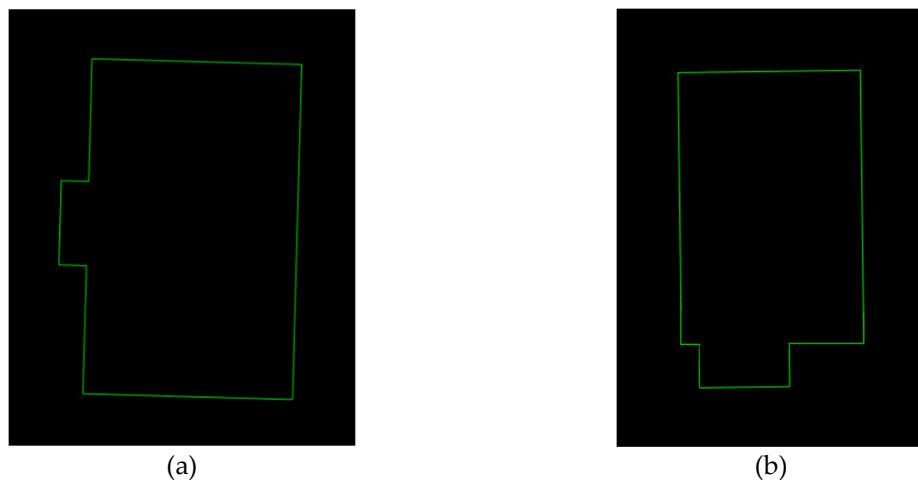


Figure 10. Border of buildings extracted by green color using ant colony algorithm. (a) Final rectangle of first simulated border and (b) final rectangle of second simulated border.

As can be seen, the proposed incorporated algorithm can extract the border of buildings with higher accuracy than the existing methods (Figures 11 and 12). However, this study showed that the proposed method can extract the border of buildings, particularly with a vertical edge. In this study, all buildings have vertical edges. Figure 11 depicts the changes and the building with no changes. In general, the changes are related to the position of points and lines on Figure 11. We determined the distance between two points horizontally. While, vertical distances are in height (elevation) and are measured along the vertical axis between points.

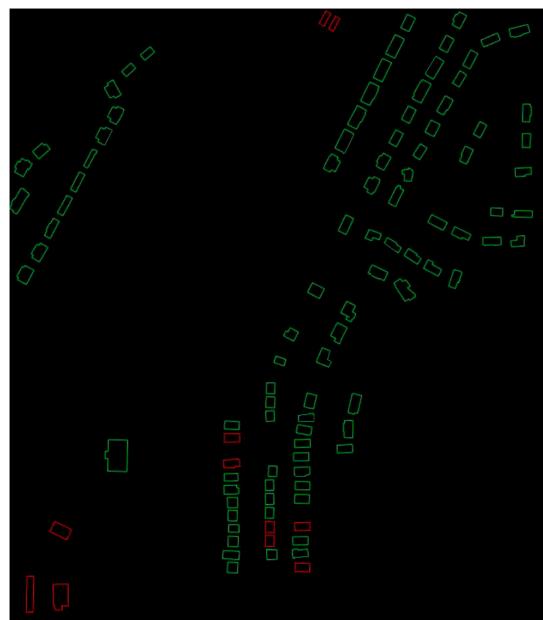


Figure 11. Unchanged buildings are shown by green color and changed buildings (newly built) are shown in red.

In Figure 11 the borders are extracted from an image with a resolution of 0.5 m. Therefore, in order to increase accuracy in border extraction, after finding point clouds of changed-buildings, an image with a resolution of 0.1 m is generated for each of them. As a result, this study reveals that the border of building can be extracted with higher accuracy by using the proposed algorithm. Figure 12 shows point clouds and the boundary of the changed buildings.

As is clear in Figures 11 and 12, there are 11 changed buildings recognized as the newly built building. For investigating the height of changed buildings, after recognizing point clouds of each building, their average height is evaluated. Then, due to the border extracted, the area of changed buildings can be calculated. Therefore, 3D change detection of buildings (horizontal and vertical) can be identified.

In Table 3 the average height and area of each changed building are shown. As can be seen from this table, the height of buildings is low and it is one of the challenges in this study area, especially in the LiDAR point clouds filtering step. Also, the largest changed building belongs to the third building that is 435.49 m² and the smallest changed building is the eleventh building that is 100.48 m².

Table 3. The average height and area of each changed building.

Area of Building (m ²)	Average Height of Building (m)	Number of Building
318.34	6.47	1
291.71	5.33	2
435.49	6.01	3
173.67	7.08	4
160.44	6.83	5
134.9	7.05	6
137.96	7.24	7
162.9	6.76	8
171.82	6.86	9
114.57	4.03	10
100.48	4.08	11

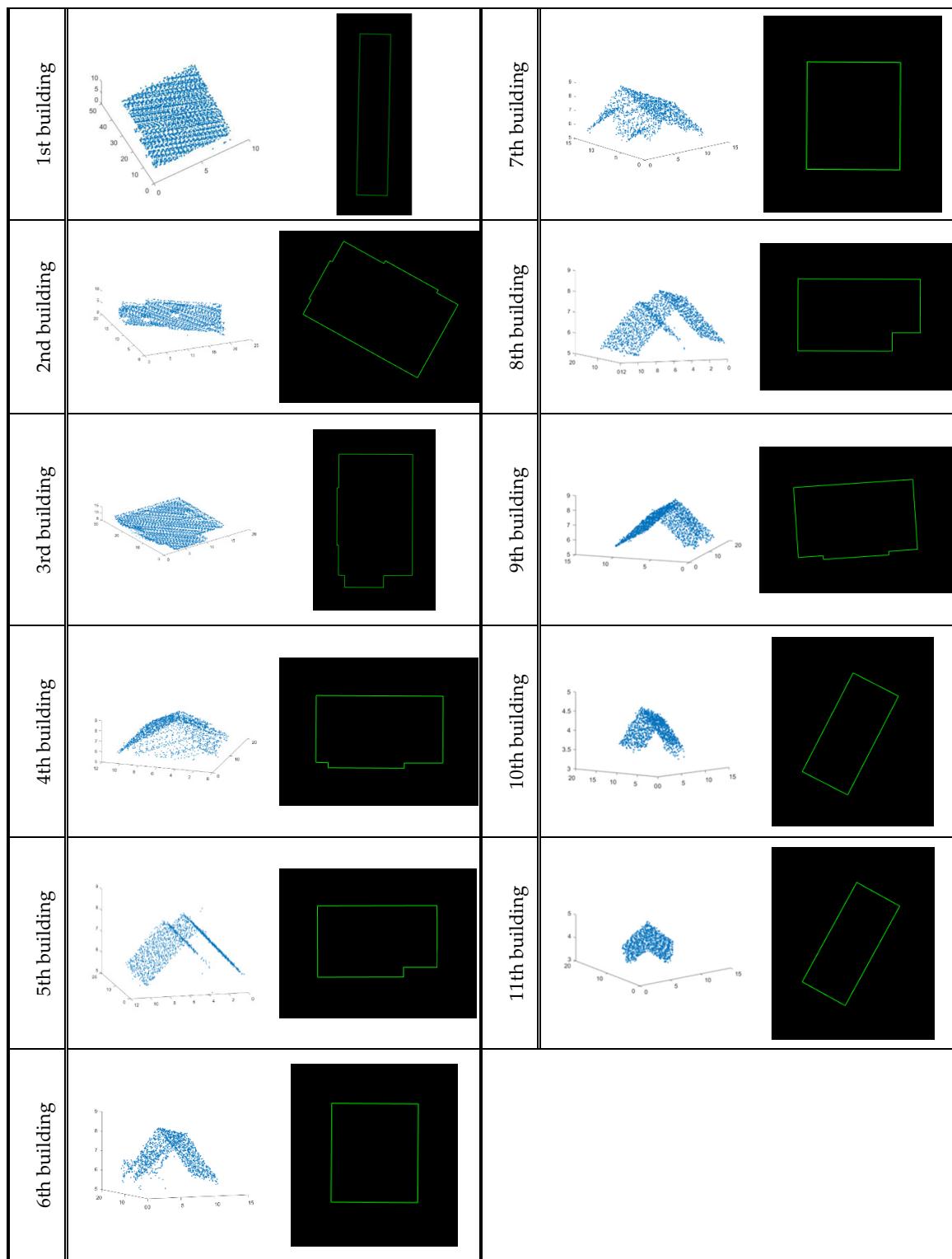


Figure 12. Point clouds and the border of changed buildings.

In addition to the aforementioned, we run the mask R-CNN on a clip image from the training datasets and two clips image from testing datasets. As the image was clipped from the DOM image, the results contained the marked roof area of the buildings. The building area was circled by bounding box and marked by individual color masks. The first result of the image (Figure 13a) reveals that the mask area is well fitted to the building area in the image when we use the image training dataset.

The other two results of images (Figure 13b,c) also fitted the building object which indicates almost 80% accuracy and the mask area of 70%. Therefore, we concluded that the mask R-CNN in the two images (Figure 13b,c) performed with an acceptable result when we apply it for small building detection.

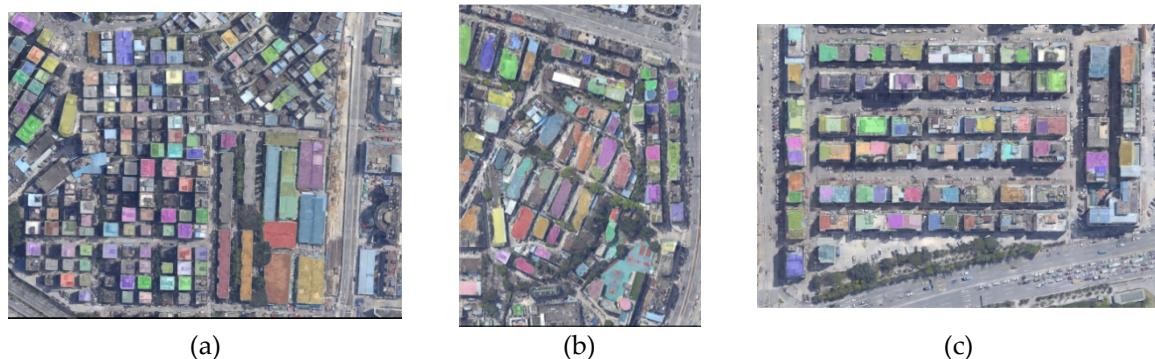


Figure 13. Bounding box and mask area of building using the mask region-based convolutional neural network (R-CNN) method for three training dataset. (a) region 1, (b) region 2, and (c) region.

5. Validation of the Proposed Method and R-CNN

While, this study recommends another set of data in different urban situation with other accuracy metrics, including geometric and positional once and not only the statistical measures to validate; however, it is worth noting that the proposed method successfully identified all of the changes in buildings accurately and no building was removed in the building extraction section. Also, for evaluating the proposed algorithm in borders extraction, the borders of buildings are extracted manually by using ArcGIS software version 10.2 and compared to the proposed method.

In a geographic information system (GIS), each region that is digitized belongs to a polygon and can be separated to the vertexes. Meanwhile, the area is one of the most important attributes of these regions and can be evaluated by coordinates according to Equation (8) [36].

$$S = \frac{1}{2} \sum_{i=1}^n x_i(y_{i+1} - y_{i-1}) \quad (8)$$

In addition, after extracting the border of buildings manually for comparing and evaluating the proposed algorithm in border extraction, this study used root mean square error (RMSE) based on Equation (9). In this equation, n is the number of buildings, S_i is the area of a building that calculated using the proposed algorithm, and S_j is the area of the building that manually calculated. In Table 4, the calculated area of buildings using the proposed method and manually is shown.

As can be seen from Table 4, the least difference for the building area is belonging to the fifth building and it is 1.21 m^2 . Also, the most significant difference is for the eighth building that is 3.83 m^2 . Meanwhile, the maximum area of building belongs to the third building and this building is one of the complex buildings that the area difference between the proposed method and ArcGIS is 2.84 m^2 . In addition, the amount of RMSE is 2.40 m^2 and, therefore, this study shows that the proposed algorithm can successfully extract the border of buildings with a higher accuracy than the existing methods.

Although in this study, a new method of extracting the border of buildings after the change detection was proposed, we assumed that the shape of buildings can simulate through some rectangles and in curved structures the accuracy of building boundary extraction may decrease. Also, an algorithm was developed for 3D change detection based on the building comparison, and it can reduce the ambiguities that are existing in methods based on the DSMs.

$$RMSE = \sqrt{\frac{\sum_{i=1}^n (S_i - S_j)^2}{n}} \quad (9)$$

Table 4. Comparison of buildings area between the results and reference data from the USA.

Area of Buildings by the Proposed Method (m^2)	Area of Buildings by ArcGIS Software (m^2)	Number of Building
318.34	315.58	1
291.71	293.15	2
435.49	438.33	3
173.67	175.59	4
160.44	161.23	5
134.9	133.29	6
137.96	139.55	7
162.9	166.73	8
171.82	168.3	9
114.57	117.11	10
100.48	102.63	11

The mask R-CNN result was validated from one image clip of training dataset and two other image clips of the testing dataset (Table 5). We also selected 11 buildings and calculated the area. As can be seen from Table 5, the lowest difference for the selected building area on the drone images is $0.84 m^2$ and belongs to the third building. Also, the most significant difference belongs to the fourth building which is depicted in Table 5 with $71.83 m^2$. Meanwhile, the highest area of building belongs to the first building with $65.17 m^2$ difference. The RMSE is 34.50 and it is very high value, reveals that the proposed method with lower RSME is a better approach.

Table 5. Comparison of buildings area between the mask R-CNN results and reference data from China.

Area of Buildings by the Mask R-CNN (m^2)	Area of Buildings by ArcGIS Software (m^2)	Number of Building
469.54	534.71	1
96.53	108.04	2
102.30	103.14	3
297.45	369.28	4
474.83	518.55	5
205.62	266.27	6
258.83	276.39	7
386.48	406.26	8
268.49	283.15	9
214.53	223.80	10
238.81	243.59	11

6. Conclusions and Recommendation

Despite most previous studies being based on differences between DSMs for change detection, this study presented an algorithm for 3D change detection based on the building comparison and considered each building separately in two different LiDAR point clouds. This algorithm can reduce the ambiguity in the change detection of buildings. The study concluded that the proposed algorithm can present a set of processes to detect and classify changes in buildings in an urban area. This study also concluded that this algorithm of border extraction of building is well suited to simulate the shape

of the building. The high performance of the filtering method causes removal of the adverse effect of the topography on building extraction. The experimental results revealed that the proposed algorithm can detect the changes in buildings in an acceptable conclusion. The type of changes to the buildings was also effectively determined. Moreover, the accuracy of the algorithm for border extraction of buildings was well and the amount of RMSE was just 2.40 m^2 as compared to R-CNN. Therefore, this algorithm using LiDAR can be suggested for use as a successful method for change detection of buildings and extraction of borders of buildings. In future work, the authors intend to detect changes in trees and ground in urban and rural areas.

In this study, we tried to compare both LiDAR and image to determine how LiDAR and the proposed method are more efficient than the R-CNN method. Therefore, we revealed that when we use LiDAR and the proposed method, we can accomplish a better result. Although the mask R-CNN is able to come up with the building boundary, it only gives a good result for small buildings and is not able to be used for 3D model. The limits of the mask R-CNN are obvious. This study concluded that the proposed method using LiDAR point clouds may be also a more effective approach than the R-CNN method using drone images. We recommend using R-CNN method on the point clouds which can be extracted from the drone images by computer vision techniques in the future study.

Author Contributions: Methodology, P.R., H.R., S.P., and S.H.; software, P.R. and H.R.; validation, S.P., S.H.; formal analysis, P.R., S.H., E.S., and S.P.; investigation, S.P.; resources, S.P., Q.Z., G.L., J.L., and Y.L.; data curation, P.R. and S.P. and S.H.; writing—original draft preparation, S.P. and P.R.; writing—review and editing, S.P.; visualization, S.P.; supervision, H.R. and S.P.; project administration, S.P.; funding acquisition, S.P.

Funding: This research received funds for publication of this research outcomes from the start up funds, Faculty of Geosciences and Environmental Engineering (FGEE), Southwest Jiaotong University (SWJTU), China.

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

References

1. Dong, P.; Zhong, R.; Yigit, A. Automated parcel-based building change detection using multitemporal airborne LiDAR data. *Surv. Land Inf. Sci.* **2018**, *77*, 5–13.
2. Vu, T.T.; Matsuoka, M.; Yamazaki, F. LIDAR-based change detection of buildings in dense urban areas. In Proceedings of the Geoscience and Remote Sensing Symposium, Anchorage, AK, USA, 20–24 September 2004.
3. Pirasteh, S.; Li, J.; Attarzadeh, I. Implementation of the damage index approach to rapid evaluation building resistance for earthquakes. *Earth Sci. Inform.* **2015**, *8*, 751–758. [[CrossRef](#)]
4. Dong, P.; Guo, H. A framework for automated assessment of post-earthquake building damage using geospatial data. *Int. J. Remote Sens.* **2012**, *33*, 81–100. [[CrossRef](#)]
5. Brunner, D.; Lemoine, G.; Bruzzone, L. Earthquake damage assessment of buildings using VHR optical and SAR imagery. *IEEE Trans. Geosci. Remote Sens.* **2010**, *48*, 2403–2420. [[CrossRef](#)]
6. Tewkesbury, A.P.; Comber, A.J.; Tate, N.J.; Lamb, A.; Fisher, P.F. A critical synthesis of remotely sensed optical image change detection techniques. *Remote Sens. Environ.* **2015**, *160*, 1–14. [[CrossRef](#)]
7. Song, D.X.; Huang, C.; Sexton, J.O.; Channan, S.; Feng, M.; Townshend, J.R. Use of Landsat and Corona data for mapping forest cover change from the mid-1960s to 2000s: Case studies from the Eastern United States and Central Brazil. *ISPRS J. Photogramm. Remote Sens.* **2015**, *103*, 81–92. [[CrossRef](#)]
8. Niederöst, M. Automated update of building information in maps using medium-scale imagery. In *Automatic Extraction of Man-Made Objects from Aerial and Space Images (III)*; Baltsavias, E., Gruen, A., Van Gool, L., Eds.; Balkema: Lisse, The Netherlands, 2001; pp. 161–170.
9. Walter, V. Object-based classification of remote sensing data for change detection. *ISPRS J. Photogramm. Remote Sens.* **2004**, *58*, 225–238. [[CrossRef](#)]
10. Knudsen, T.; Olsen, B.P. Automated change detection for updates of digital map databases. *Photogramm. Eng. Remote Sens.* **2003**, *69*, 1289–1296. [[CrossRef](#)]
11. Vu, T.T.; Matsuoka, M.; Yamazaki, F. (Eds.) LIDAR for updating 3D information in building database. In Proceeding of the ASIA Conference on Earthquake Engineering (ACEE), Vancouver, BC, Canada, 1–6 August 2004.

12. Qin, R.; Tian, J.; Reinartz, P. 3D change detection—approaches and applications. *ISPRS J. Photogramm. Remote Sens.* **2016**, *122*, 41–56. [[CrossRef](#)]
13. Radovic, M.; Adarkwa, O.; Wang, Q. Object Recognition in Aerial Images Using Convolutional Neural Networks. *J. Imaging* **2017**, *3*, 21. [[CrossRef](#)]
14. Uijlings, J.R.; Van de Sande, K.E.; Gevers, T.; Smeulders, A.W. Selective search for object recognition. *Int. J. Comput. Vis.* **2013**, *104*, 154–171. [[CrossRef](#)]
15. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014.
16. He, K.; Zhang, X.; Ren, S.; Sun, J. Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *37*, 1904–1916. [[CrossRef](#)] [[PubMed](#)]
17. Girshick, R. Fast R-CNN. *arXiv* **2015**, arXiv:1504.08083.
18. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 1137–1149. [[CrossRef](#)] [[PubMed](#)]
19. Murakami, H.; Nakagawa, K.; Hasegawa, H.; Shibata, T.; Iwanami, E. Change detection of buildings using an airborne laser scanner. *ISPRS J. Photogramm. Remote Sens.* **1999**, *54*, 148–152. [[CrossRef](#)]
20. Vögtle, T.; Steinle, E. Detection and recognition of changes in building geometry derived from multitemporal laserscanning data. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2004**, *35*, 428–433.
21. Teo, T.A.; Shih, T.Y. Lidar-based change detection and change-type determination in urban areas. *Int. J. Remote Sens.* **2013**, *34*, 968–981. [[CrossRef](#)]
22. Xu, S.; Vosselman, G.; Oude Elberink, S. Detection and classification of changes in buildings from airborne laser scanning data. *Remote Sens.* **2015**, *7*, 17051–17076. [[CrossRef](#)]
23. Xu, H.; Cheng, L.; Li, M.; Chen, Y.; Zhong, L. Using octrees to detect changes to buildings and trees in the urban environment from airborne LiDAR data. *Remote Sens.* **2015**, *7*, 9682–9704. [[CrossRef](#)]
24. Tran, T.H.G.; Ressl, C.; Pfeifer, N. Integrated change detection and classification in urban areas based on airborne laser scanning point clouds. *Sensors* **2018**, *18*, 448. [[CrossRef](#)] [[PubMed](#)]
25. Stal, C.; Tack, F.; De Maeyer, P.; De Wulf, A.; Goossens, R. Airborne photogrammetry and lidar for DSM extraction and 3D change detection over an urban area—a comparative study. *Int. J. Remote Sens.* **2013**, *34*, 1087–1110. [[CrossRef](#)]
26. NSF. Available online: <http://opentopo.sdsc.edu/datasets?listAll=true> (accessed on 20 April 2019).
27. Meng, X.; Wang, L.; Silván-Cárdenas, J.L.; Currit, N. A multi-directional ground filtering algorithm for airborne LIDAR. *ISPRS J. Photogramm. Remote Sens.* **2009**, *64*, 117–124. [[CrossRef](#)]
28. Li, Y.; Wu, H.; Xu, H.; An, R.; Xu, J.; He, Q. A gradient-constrained morphological filtering algorithm for airborne LiDAR. *Opt. Laser Technol.* **2013**, *54*, 288–296. [[CrossRef](#)]
29. Rashidi, P.; Rastiveis, H. Ground Filtering LiDAR Data Based on Multi-Scale Analysis of Height Difference Threshold. In Proceedings of the ISPRS Conferences of GI Research, SMPR and EOEC, Tehran, Iran, 7–10 October 2017.
30. Pirasteh, S.; Li, J.; Chapman, M. Use of LiDAR-derived DEM and a stream length-gradient index approach to investigation of landslides in Zagros Mountains, Iran. *Geocarto Int.* **2018**, *33*, 912–926. [[CrossRef](#)]
31. Kobler, A.; Pfeifer, N.; Ogrinc, P.; Todorovski, L.; Oštir, K.; Džeroski, S. Repetitive interpolation: A robust algorithm for DTM generation from Aerial Laser Scanner Data in forested terrain. *Remote Sens. Environ.* **2007**, *108*, 9–23. [[CrossRef](#)]
32. Zhang, J.; Lin, X. Filtering airborne LiDAR data by embedding smoothness-constrained segmentation in progressive TIN densification. *ISPRS J. Photogramm. Remote Sens.* **2013**, *81*, 44–59. [[CrossRef](#)]
33. St-Onge, B.; Vepakomma, U. Assessing forest gap dynamics and growth using multi-temporal laser-scanner data. *Power* **2004**, *140*, 173–178.
34. Vastaranta, M.; Korpela, I.; Uotila, A.; Hovi, A.; Holopainen, M. Mapping of snow-damaged trees based on bitemporal airborne LiDAR data. *Eur. J. For. Res.* **2012**, *131*, 1217–1228. [[CrossRef](#)]
35. Li, Y.; Yong, B.; Wu, H.; An, R.; Xu, H. An improved top-hat filter with sloped brim for extracting ground points from airborne lidar point clouds. *Remote Sens.* **2014**, *6*, 12885–12908. [[CrossRef](#)]
36. Rashidi, P.; Rastiveis, H. Extraction of ground points from LiDAR data based on slope and progressive window thresholding (SPWT). *Earth Observ. Geomat. Eng.* **2018**, *2*, 36–44.

37. Yang, X.S.; He, X. Firefly algorithm: Recent advances and applications. *Int. J. Swarm Intell.* **2013**, *1*, 36–50. [[CrossRef](#)]
38. Hoffman, K.L.; Padberg, M.; Rinaldi, G. Traveling salesman problem. In *Encyclopedia of Operations Research and Management Science*; Springer: Berlin, Germany, 2013; pp. 1573–1578.
39. Dorigo, M.; Gambardella, L.M. Ant colony system: A cooperative learning approach to the traveling salesman problem. *IEEE Trans. Evol. Comput.* **1997**, *1*, 53–66. [[CrossRef](#)]
40. Mazzeo, S.; Loiseau, I. An ant colony algorithm for the capacitated vehicle routing. *Electron. Notes Discret. Math.* **2004**, *18*, 181–186. [[CrossRef](#)]
41. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask R-CNN. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2961–2969.



© 2019 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).