

Research Article

Segmentation of LiDAR Data Using Multilevel Cube Code

So-Young Park¹, Dae Geon Lee ², Eun Jin Yoo², and Dong-Cheon Lee ²

¹ICT Business Unit, KT Hitel Co., Seoul 07071, Republic of Korea

²Department of Environment, Energy and Geoinformatics, Sejong University, Seoul 05006, Republic of Korea

Correspondence should be addressed to Dong-Cheon Lee; dcleee@sejong.ac.kr

Received 24 January 2019; Accepted 1 April 2019; Published 17 April 2019

Guest Editor: Sang-Hoon Hong

Copyright © 2019 So-Young Park et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Light detection and ranging (LiDAR) data collected from airborne laser scanning systems are one of the major sources of spatial data. Airborne laser scanning systems have the capacity for rapid and direct acquisition of accurate 3D coordinates. Use of LiDAR data is increasing in various applications, such as topographic mapping, building and city modeling, biomass measurement, and disaster management. Segmentation is a crucial process in the extraction of meaningful information for applications such as 3D object modeling and surface reconstruction. Most LiDAR processing schemes are based on digital image processing and computer vision algorithms. This paper introduces a shape descriptor method for segmenting LiDAR point clouds using a “multilevel cube code” that is an extension of the 2D chain code to 3D space. The cube operator segments point clouds into roof surface patches, including superstructures, removes unnecessary objects, detects the boundaries of buildings, and determines model key points for building modeling. Both real and simulated LiDAR data were used to verify the proposed approach. The experiments demonstrated the feasibility of the method for segmenting LiDAR data from buildings with a wide range of roof types. The method was found to segment point cloud data effectively.

1. Introduction

Automatic building modeling using light detection and ranging (LiDAR) data is a challenging and complex task because the point clouds obtained from airborne laser scanning systems, which consist of unstructured 3D coordinates, do not provide the topological and semantic information that allow building surfaces to be properly segmented. Point density, complicated roof shapes, occlusion, and unwanted features further complicate the segmentation necessary for building modeling [1]. The ultimate goal, utilizing geospatial sensor data (e.g., aerial and satellite images, laser scanner data, and radar data), is the automatic reconstruction of real-world objects, and the core challenge of automatic processing is surface patch segmentation. Although the task is challenging and remains unresolved, considerable research has been devoted to its solution. Many of the methods are ad hoc or solve only partial problems, and research continues. Explicit description of object surfaces, with meaningful information, is crucial [2]. Nowadays, segmentation of the point cloud data, from unmanned aerial vehicles or drones, is increasingly necessary

for object modeling because much of the image processing is based on point clouds that are generated by multiview stereo matching.

Segmentation for object modeling is the central issue in effective processing of LiDAR point clouds. Methods for segmentation of point clouds are categorized as (1) region growing, (2) edge-based approach, (3) model fitting, (4) parameter-domain clustering methods, and (5) hybrid methods. Hybrid methods include combining data-driven and model-driven approaches or fusing various datasets (e.g., LiDAR data with images or maps) [3–7]. Maas and Vosselman [8] proposed a building modeling method based on moment invariant parameters. Vosselman and Dijkman [9] developed a 3D building modeling approach involving segmentation and grouping of LiDAR data using the Hough transform and building planes that provide building boundaries. Rottensteiner and Briesse [10] described procedures for automatic building detection, segmentation, and modeling, especially curved roofs. Miliareisis and Kokkas [11] applied a region growing technique for segmentation, and Sampath and Shan [12] used a modified convex hull method for modeling

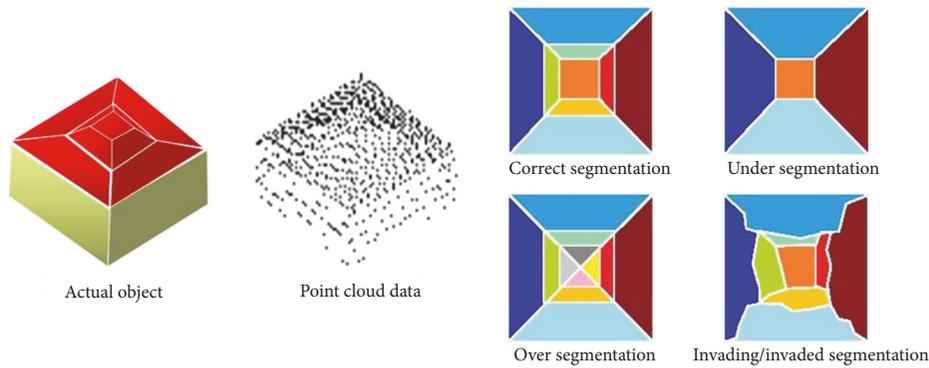


FIGURE 1: Types of erroneous segmentations.

various shapes of buildings. Sampath and Shan [4] segmented and modeled complicated buildings from airborne LiDAR data based on polyhedral models. They detected nonplanar points after using eigen-analysis for robust segmentation of roof surfaces. Huang et al. [13] proposed a top-down approach to reconstruct roofs based on stochastic modeling while most modeling methods are bottom-up. In a recent building reconstruction and modeling study, Chen et al. [14] presented a building reconstruction process for clustering roof primitives according to their topological consistency. The method is based on a probability density clustering algorithm. Gilani et al. [1] introduced a robust method for detecting buildings and segmenting various roof types using principal component analysis and region growing with an optimal seed selection scheme. Loutfia et al. [15] proposed using orthoimages with LiDAR data for 3D city modeling based on the fuzzy *c*-means clustering algorithm. Yi et al. [16] presented a building reconstruction method that decomposed point clouds into an individual building based on a divide-and-conquer strategy.

Figure 1 depicts the typical problems that occur during segmentation. The problems may be classified as under-, over-, and invading/invaded segmentation [1, 17, 18].

Many segmentation methods have been developed; however, a standard method is not yet available. The key issue in segmenting LiDAR data with irregularly distributed point clouds is how to measure similarity. The motivation of the proposed method is based on the following. (1) A surface patch to be treated as a single segment has the same geometric properties, such as slope and orientation, in 3D space. (2) Man-made structures including buildings can be decomposed into surface patches having simple geometric shapes including planar (i.e., horizontal, vertical, or sloped plane) and smoothly curved surfaces (i.e., hemisphere or half-cylinder). In other words, complicated roof shapes can be segmented into geometrically simple surfaces.

One goal of segmentation is to identify and group points derived from the same surface patches of building roofs. The proposed method is based on the shape descriptor concept. The slope is an essential factor in describing the geometric characteristics of the surfaces. The slope of each surface provides information about the orientation of the surfaces and indirectly their shape. Consequently, the ability

to group cloud points whose surfaces have the same slope is central to segmenting the LiDAR data. An approach based on 3D chain code is proposed here. Chain code was originally developed to describe the shapes of objects in digital images by extracting object boundaries and describing them using eight orientations and the lengths of the resulting line segments [19]. To apply the chain code operator to 3D point cloud data, 2D chain code analysis was extended to 3D space (called “cube code” here). Segmentation is complicated by noise, meaningless and unnecessary objects, in the point cloud. Therefore, such data should be removed as much as possible. In particular, vegetation close to buildings inhibits segmentation and may produce incorrect building models. The proposed method removes such undesirable data and also separates ground and nonground features.

The cube code operator is able to extract meaningful information from the data through a series of processes for removing unnecessary point clouds (e.g., trees and vegetation), ground/nonground feature separation, building extraction, surface patch segmentation, and model key point (MKP) detection. The MKPs are important points, such as corners, essential for building modeling. A key feature of the proposed method is the implementation of adaptive 3D chain code within a multicube that has been divided into subcubes. An adaptive multilevel approach makes it possible to segment complex roofs with a high level of detail (LoD). Object recognition should be invariant under geometric transformations, including scale, rotation, and shift. Invariance was achieved by applying the operator from different directions. Then, roof surface patch segmentation was performed by combining the results from each direction. Finally, by utilizing information from the chain codes, the MKPs were extracted from the segmented data.

2. Methods

Two key issues in effective segmentation are defining adjacency criteria among points and defining grouping or clustering criteria [20]. The proposed segmentation method is based on a 3D shape descriptor to recognize objects. The basic idea is to group the same cube codes together as a segmented region. The cube operator can segment point clouds by classifying coded surface slopes through an extension of the 2D

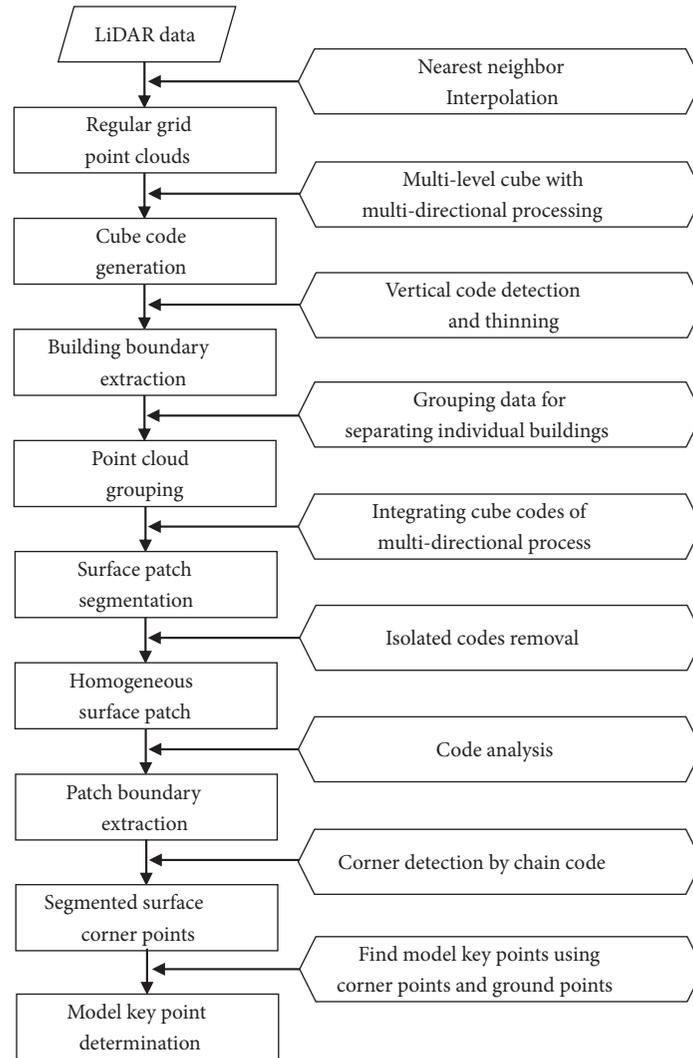


FIGURE 2: Workflow of the proposed method.

chain code to 3D space. The method requires interpolation of the point clouds to rearrange the irregularly spaced LiDAR points into regular grid cells to meet the input requirements of the cube code operator. The interpolation method and sample size (i.e., GSD: ground sample distance) applied to the LiDAR data are critical because these affect the LoD with which features can be identified [21, 22]. The interpolation process is not the main focus of this study and the nearest-neighbor interpolation was applied. Figure 2 outlines the proposed method.

2.1. Cube Code Operator. Chain code is a boundary encoding method proposed by Freeman [19]. It describes a sequence of unit line segments, with a limited set of possible discretized directions, in 2D space. These boundary elements can be segmented into line segments based on directional changes. The directions and lengths of line segments along an object boundary identify its shape. Therefore, chain code, as a shape descriptor, has been used for object recognition. We have

extended the chain code method to 3D space to segment point clouds by using a corresponding cube operator. The size of the cube is $3 \times 3 \times 3$ (i.e., three layers with 3×3 cells for each layer), giving 27 cubic cells or voxels, and the cube code operator looks like Rubik's cube, as shown in Figure 3. There are 26 possible directions from the center voxel of the cube to an adjacent point. Sánchez-Cruz et al. [23] used such 3D chain code to represent 3D discrete curves. In this paper, the 3D version of the chain code is called "cube code."

The drawbacks of the original chain code are (1) the limited number of discrete directions (generally eight directions), (2) being variant under rotation, and (3) being sensitive to noise. As a result, the shapes of the objects may not be accurately described. Clearly, such problematic characteristics of the chain code can also occur in the cube code, leading to incorrect segmentation of the point cloud data.

In our work, to improve segmentation, these problems were resolved as follows:

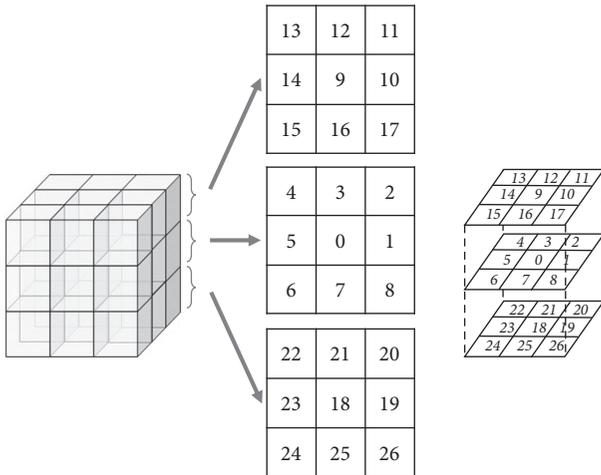


FIGURE 3: Cube code with 26 directions in 3D space.

- (1) Creation of an adaptive multilevel cube.
- (2) Processing with different directions, and combining results, to make these rotations invariant.
- (3) Refinement by grouping segmented surfaces to reduce noise effects.

When applying the chain code approach to an image, the boundaries of the objects are extracted and chain codes are created along the boundaries. Cube codes, on the other hand, are created at each point of the point cloud. The direction assigned to each chain code corresponds to the surface slope at that point in the cube code. Surfaces are then segmented on the basis of the slope information in the cube code. In addition, it is possible to remove unwanted objects (e.g., trees and other vegetation) and to extract model key features (e.g., boundaries of the segmented surfaces and corner points) of the buildings by analyzing the cube code. Each step is described in detail below.

2.2. Multilevel Cube Code. Because cube code, in the same way as chain code, has a limited number of discrete directions, details of the objects may not be precisely represented and could result in undersegmentation. To solve this problem, a multilevel cube, consisting of $3 \times 3 \times 3$ subcubes, was introduced to the process. Within each subcube the same $3 \times 3 \times 3$ voxel division applies, again with 26 possible directions. The multilevel cube enables adaptive determination of surface slope, recognition of the surface shape, and thus appropriate segmentation. As shown in Figure 4(a), a cube without subcubes could not distinguish a sloped surface from a flat surface if the slope is low because of the limited number of discrete directions. Therefore, a cube having a uniform size could not work properly to segment point clouds generated by roofs with diverse shapes.

The subcubes in a multilevel cube are adaptively generated according to the slope between points. If the slopes between adjacent points are smaller than a threshold value (ϵ), subcubes are generated in the voxels of the original cube. The multilevel cube allows generation of building models

with higher LoD. Because the subcubes are noise sensitive, and the slopes of the surfaces could become exaggerated, results from both the original and subcube analyses are combined for more accurate segmentation. The multicube process is illustrated in Figure 4. The flowchart in Figure 5 shows the criteria that determine whether subcubes are created.

2.3. Multidirectional Processing. Another important feature of the proposed method is multidirectional processing. As shown in Figure 6 for a gable roof example, the slope of a roof surface in 3D space can be decomposed into X- and Y-directions. Alternatively, the surface can be represented by slope and aspect (i.e., the orientation of the slope). It is obvious that surfaces with different slopes could not be distinguished by one directional processing. Therefore, the cube code operator was applied to different directions (i.e., S, E, SE, and NE) and slope components for each direction were determined as shown in Figure 7. Because surfaces can be arbitrarily oriented in the object coordinate system (e.g., building “C” in Figure 6), and as the roof shapes of some buildings are complex, applying the cube operator in several different directions can provide more robust and detailed slope information.

The cube code is not intrinsically rotation invariant. Therefore, if the cube operator is applied in only one direction, some adjacent surfaces of similar orientation might not be distinguished from each other, leading to undersegmentation. To reduce this risk, the cube code is applied in different directions (i.e., S, E, SE, and NE) and the results from each direction are combined. In this work, the coordinate system of the object space was based on the map coordinate system with Easting (E), Northing (N), and elevation instead of a local Cartesian system (i.e., X, Y, and Z coordinates). In this paper, X, Y, and Z denote E, N, and elevation, respectively.

2.4. Removal of Unnecessary Objects and Building Boundary Extraction. Noise in the point cloud should be removed, or reduced as much as possible. Trees, other vegetation, cars, and small objects that are not part of buildings should be excluded from the building analysis data. Such unnecessary objects can be identified and removed by analyzing cube codes. As shown in Figure 3, the vertical direction components of the cube code (i.e., 9 and 18) indicate possible boundaries of buildings and boundaries of rooftop superstructures having vertical sidewalls (e.g., an attic or chimney). However, noise or other objects might suggest vertical components in the cube code. The major purpose of this step is to identify vertical components arising from trees or vegetation areas and remove such unnecessary objects. The vertical components derived from trees and other vegetation areas, and their associated cube codes, tend to be more randomly distributed than those from building surfaces. Variations (as estimated by their standard deviation) in the cube codes arising from such areas would be relatively large. In addition, the vertical components would lack meaningful connectivity with each other and, even if polygon forming, tend to enclose smaller areas. The method is thus able to separate vertical components of

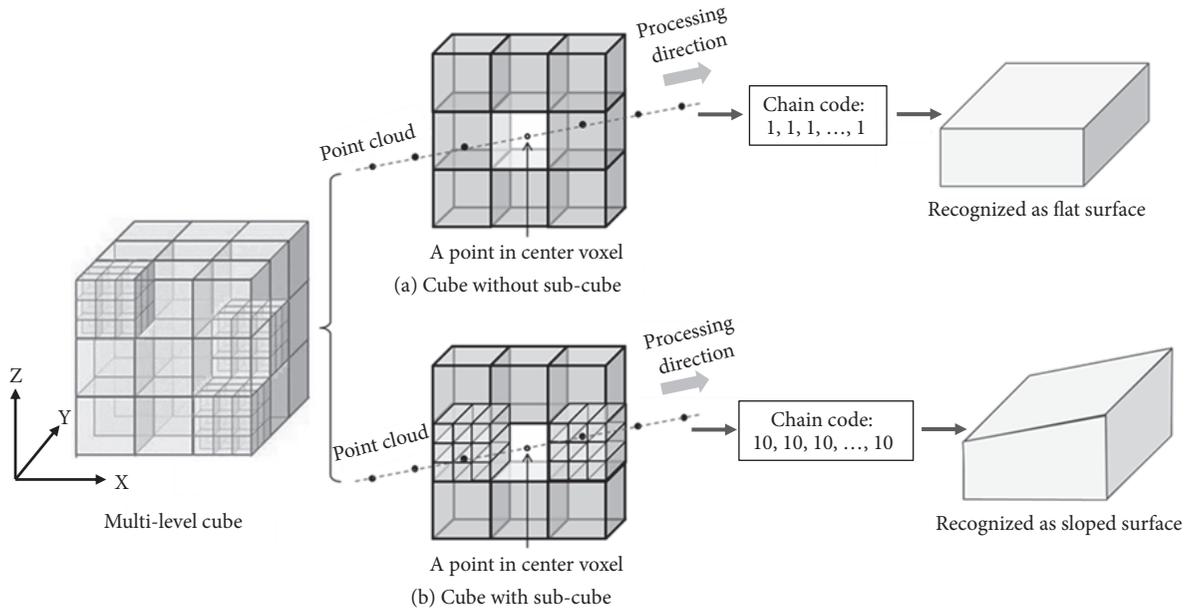


FIGURE 4: Demonstration of a multilevel cube.

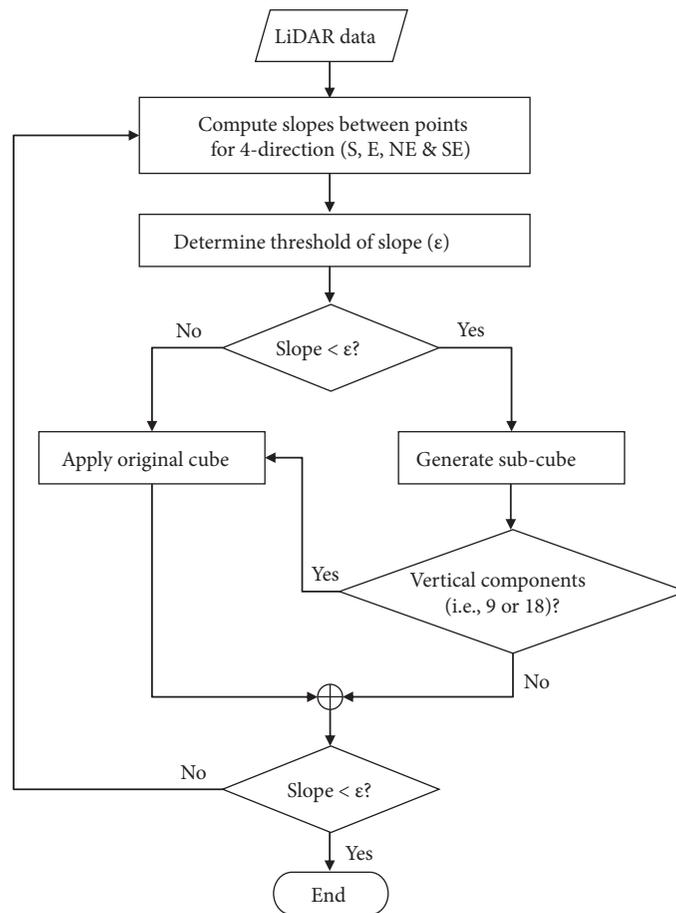


FIGURE 5: Flowchart of multilevel cube process.

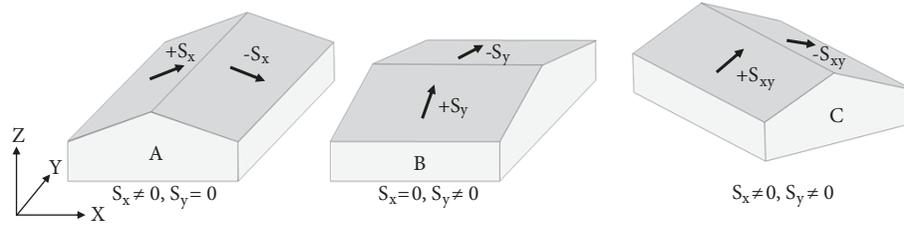


FIGURE 6: 3D slope vectors of surfaces for different surface orientations. S_x , S_y , and S_{xy} denote the slopes in the X, Y, and XY combined directions, respectively. + and - indicate ascending and descending slopes, respectively.

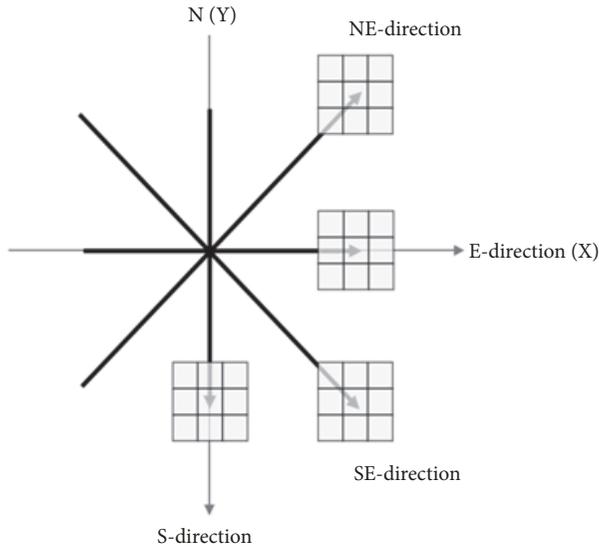


FIGURE 7: Processing directions of the cube code operator.

the vegetation areas from those of the buildings based on these characteristics (regularity, closeness, connectivity, and polygon size) of the cube codes.

Figure 8 shows the procedure for removing vegetation areas and extracting building boundaries. Figure 9 shows an example. The vertical components of the vegetation area are irregularly distributed as shown in Figure 9(b). After applying dilation, erosion, and thinning processes, the vegetation areas become irregular linear features, not closed features like the buildings (Figure 9(c)), and can be filtered out (Figure 9(d)).

2.5. Segmentation Scheme. Segmentation is a process to partition data into meaningful and homogeneous regions. The adjacent regions of a segmentation should have significant differences and spatially accurate boundaries [24, 25]. No matter how complex a building roof is, it consists of several simple-shaped unit surfaces having common geometric characteristics. The shape of a surface can be determined from its slopes. Therefore, surface slopes can distinguish surface units from each other and allow shape identification, as shown in Figure 10 and discussed further in Section 2.7.

The cube operator converts the slopes between adjacent points into codes and performs segmentation efficiently by

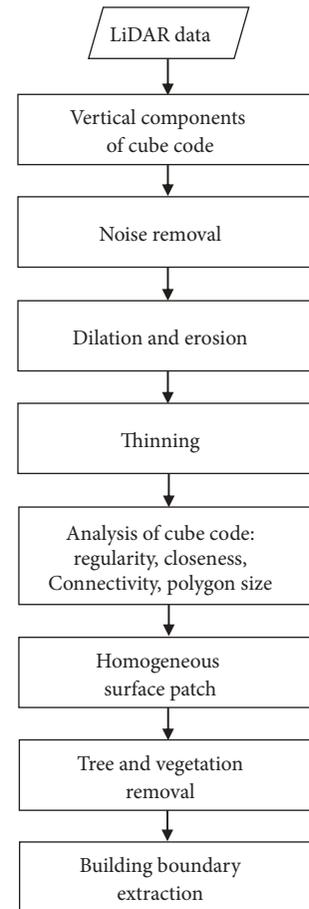


FIGURE 8: The procedure for removing vegetation areas and extracting building boundaries.

grouping the same codes. As described above, subcubes are adaptively generated based on the slopes between points allowing detailed segmentation (i.e., high LoD). In addition, cube codes generated from four directions are integrated as shown in Figure 11.

2.6. Determination of Model Key Points. The cube operator detects the corner points of the segmented surfaces, which can be used to model roofs. Codes are generated along the boundaries of the segmented regions and these are projected to the horizontal plane (i.e., X-Y plane). Then,

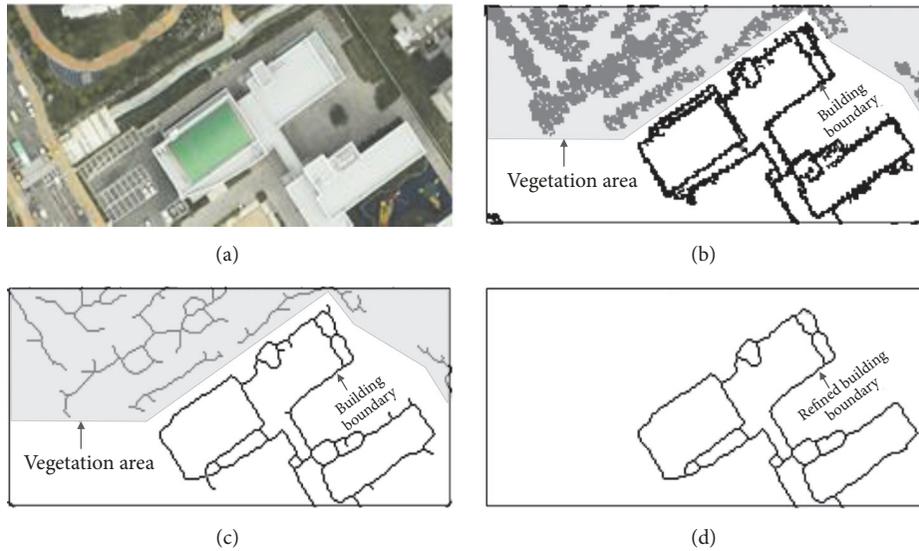


FIGURE 9: Example of unnecessary object and building boundary extraction. (a) Aerial image of test area. (b) Vertical components of cube code. (c) Thinning vertical components. (d) Result of trees and vegetation removing.

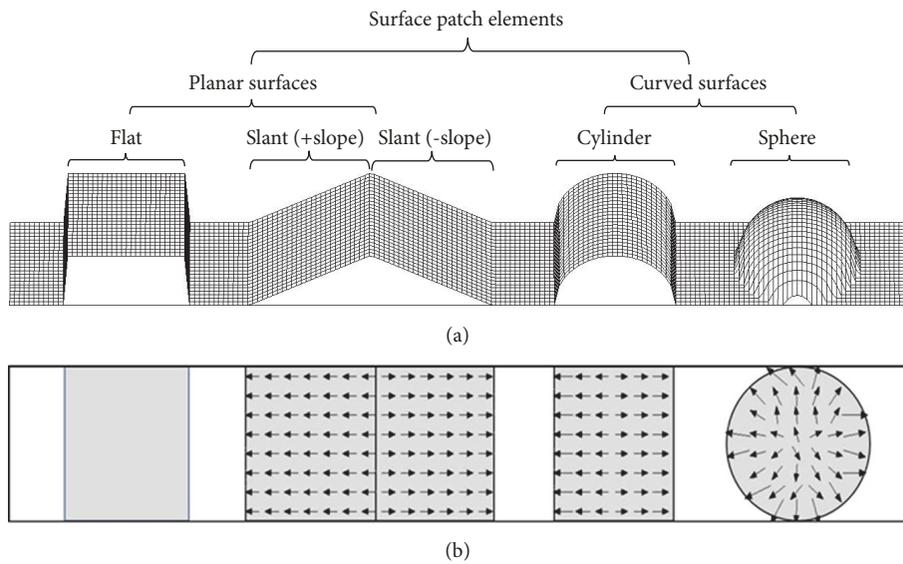


FIGURE 10: Typical shapes of roof and slope vectors. (a) Surface shape. (b) Slope vectors.

the middle layer of the cube (i.e., a conventional 2D chain code with eight directions) is used to detect the corner points. Locations with an abrupt change in direction are considered as the corner points. However, most of the segmentation boundaries, between corner points, are not straight lines but rather consist of irregular line segments because of noise in the point cloud and imperfections in the segmentation. Information about the direction and length of the boundary chain codes is used to prevent inappropriate points from being detected as corners. Each boundary between corner points should be represented by several identical codes. Hence, a regularization process that generates a common code along true boundary segments is required. If the code changes over a short distance, then

the points may not be true corners. Figure 12 illustrates the regularization process to detect corner points using chain code, and Figure 13 shows an example of segmentation and corner point detection using simulated LiDAR data.

However, the corner points of the segmented surfaces might not be the appropriate MKPs of the buildings because the LiDAR points on the roofs may be inside the real building boundaries, as shown in Figure 14. The actual building boundary is located somewhere between the segmented surface boundary and the nearest ground points. A reasonable estimation of the building boundaries is midway between the points on the segmented surface boundary and the nearest ground points.

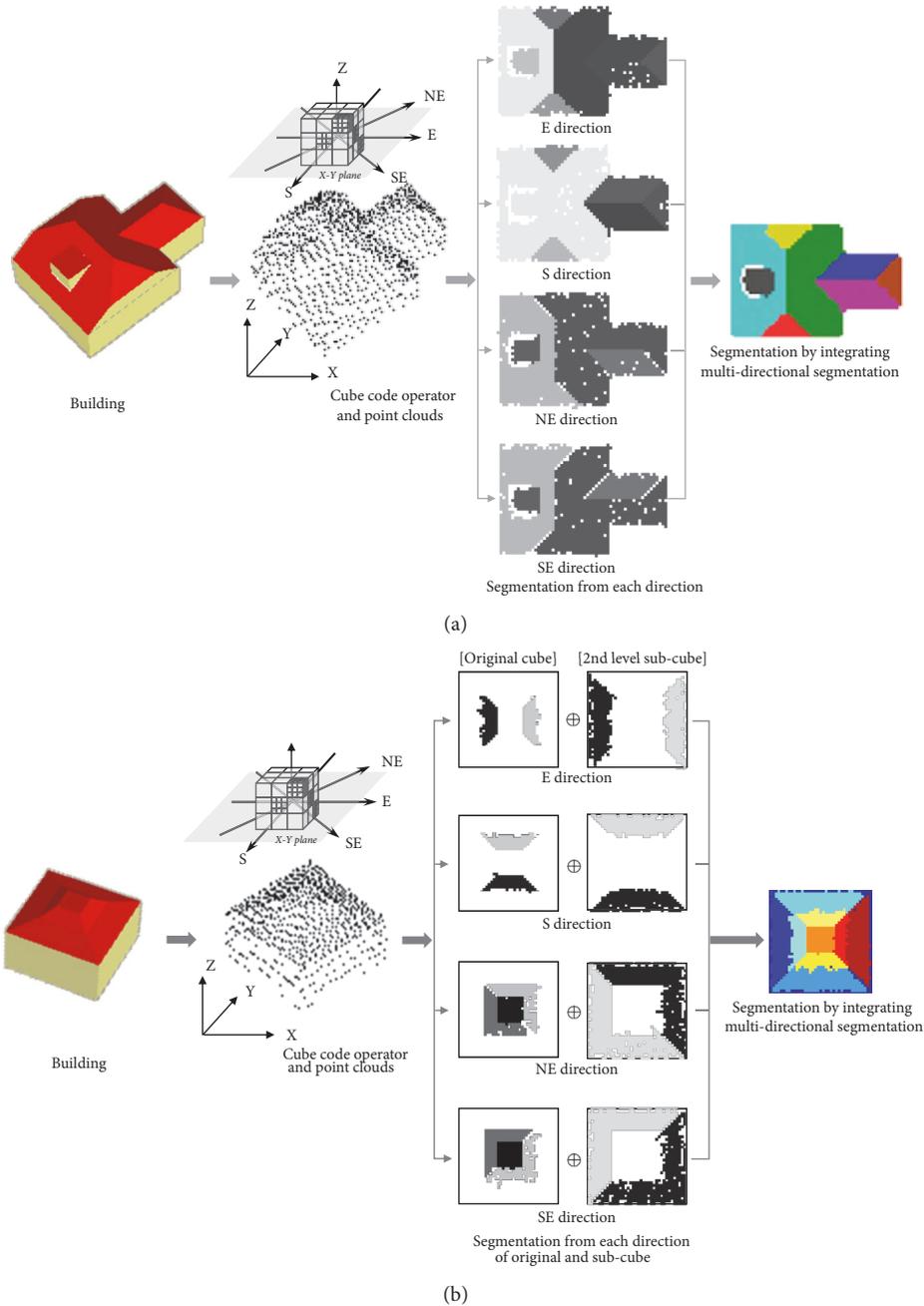


FIGURE 11: Segmentation from multilevel and multidirectional processes. (a) One-level cube. (b) Two-level cube.

The original LiDAR points may be used instead of the interpolated points to determine the MKPs more accurately, as follows:

- (1) Set search areas around corner points of segmented regions.
- (2) Compute midpoints between each corner point and the nearest point on the ground.
- (3) Determine the midpoints as MKPs.

Figure 14 shows corner points from a segmented surface and the estimated MKPs of the building. MKPs, in the remainder

of this paper, refer to these interpolated midpoints on the outermost building boundaries.

2.7. Curved Surface Modeling. We propose an idea for modeling the curved surfaces as illustrated in Figure 15. When the cube code is applied to curved surfaces such as a dome (i.e., hemisphere) or arch (i.e., half-cylinder) roof, the surfaces are segmented into several patches because the slope of the curved surfaces varies from point to point. On the other hand, planar surfaces have a constant slope, as shown in Figure 16. The hemispherical and half-cylindrical surfaces can be

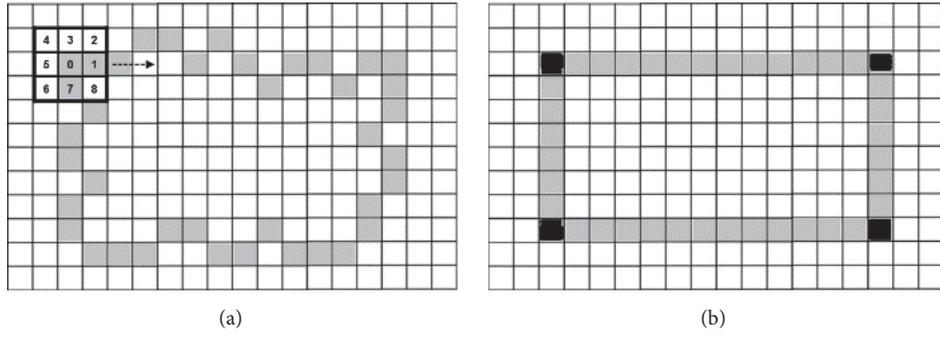


FIGURE 12: Regularization of a boundary for corner detection. (a) Segmented region boundary. (b) Regularized boundary and corners.

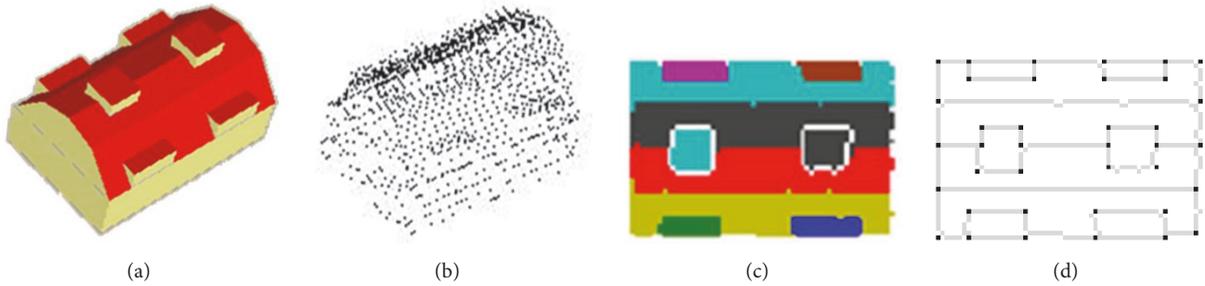


FIGURE 13: Example of corner point detection. (a) Building model. (b) Point clouds. (c) Segmentation. (d) Corner points.

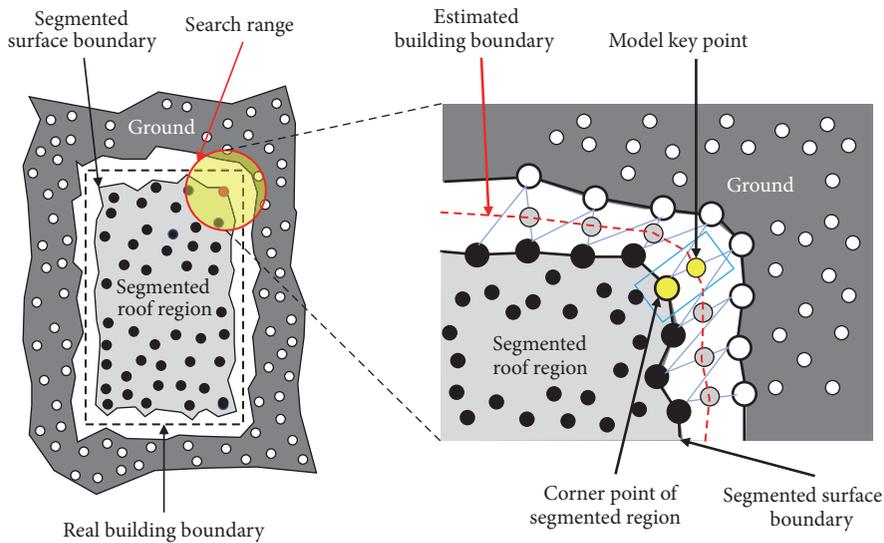


FIGURE 14: Scheme for determining MKPs.

distinguished from each other by analyzing the characteristics of the slope vector patterns. The slope vectors of the dome are uniformly distributed in all directions, while those of the half-cylinder are distributed in two dominant directions (i.e., the numbers in parentheses in Figure 15 indicate the number of slope vectors in each direction). These distribution patterns are rotation invariant.

Curved surfaces can be modeled by fitting the points on the surfaces with appropriate mathematical functions after determination of the characteristics of the slope vectors (i.e.,

the dome and half-cylindrical surfaces can be represented by sphere and cylinder equations, respectively). The parameters of the equations can be determined by least-square fitting of the points on the surfaces within the corresponding equations. The sphere and cylinder equations are nonlinear. Solving for the unknown parameters usually requires linearization of the equations, with initial approximations of the parameters followed by iteration toward a required level of the accuracy. A direct method that does not require linearization, initial approximations, and iteration is suggested here

Roof type	Roof model	Point clouds	Segmentation	Slope vectors	Distribution of slope vectors
Dome					
Half-cylinder					

FIGURE 15: Segmentations and analysis of curved surfaces.

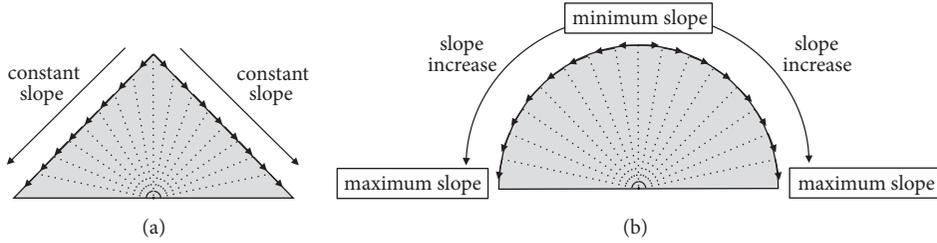


FIGURE 16: Slopes of planar and curved surfaces. (a) Planar surface with constant slope. (b) Curved surface with varying slope.

for mathematical representation of the curved surfaces. The standard equation of the sphere is

$$(X - X_o)^2 + (Y - Y_o)^2 + (Z - Z_o)^2 = r^2 \quad (1)$$

where (X, Y, Z) are coordinates of the point clouds, (X_o, Y_o, Z_o) are center coordinates, and r denotes the radius of the sphere. By introducing new parameters (i.e., $p_1 = -2X_o$, $p_2 = -2Y_o$, $p_3 = -2Z_o$, and $p_4 = X_o^2 + Y_o^2 + Z_o^2 - r^2$), the standard equation is rewritten as

$$X^2 + Y^2 + Z^2 + p_1X + p_2Y + p_3Z + p_4 = 0 \quad (2)$$

The observation equation to solve for the unknown parameters (i.e., p_1, p_2, p_3 , and p_4) by the least-square method is expressed as

$$\begin{bmatrix} -(X_1^2 + Y_1^2 + Z_1^2) \\ -(X_2^2 + Y_2^2 + Z_2^2) \\ \vdots \\ -(X_n^2 + Y_n^2 + Z_n^2) \end{bmatrix} \quad (3)$$

$$= \begin{bmatrix} X_1 & Y_1 & Z_1 & 1 \\ X_2 & Y_2 & Z_2 & 1 \\ \vdots & \vdots & \vdots & \vdots \\ X_n & Y_n & Z_n & 1 \end{bmatrix} \cdot \begin{bmatrix} p_1 \\ p_2 \\ p_3 \\ p_4 \end{bmatrix} + \begin{bmatrix} e_1 \\ e_2 \\ \vdots \\ e_n \end{bmatrix}$$

$$\bar{Y} = A \cdot \hat{\xi} + \bar{e} \quad (4)$$

$$\hat{\xi} = (A^T A)^{-1} \cdot (A^T \bar{Y}) \quad (5)$$

where \bar{Y} represents the observation vector, A is a design matrix, $\hat{\xi}$ is an estimation of the unknown parameter vector, and \bar{e} is the error vector. Finally, the center and radius of the sphere can be computed as follows:

$$\begin{aligned} X_o &= -\frac{p_1}{2}, \\ Y_o &= -\frac{p_2}{2}, \end{aligned} \quad (6)$$

$$Z_o = -\frac{p_3}{2}$$

$$r = \sqrt{\left(\frac{p_1}{2}\right)^2 + \left(\frac{p_2}{2}\right)^2 + \left(\frac{p_3}{2}\right)^2 - p_4} \quad (7)$$

For an arbitrarily oriented cylindrical surface, the key parameters are basically two orientation angles (i.e., horizontal and vertical angles) of the axis, the center, and the radius of the cylinder. The orientation angles can be determined by linear regression in 3D space. The center coordinates and radius can be estimated directly in a similar way to the sphere surface.

3. Experimental Results and Analysis

3.1. Description of Datasets. Simulated and actual LiDAR datasets were tested to validate the proposed method. It is not easy to obtain LiDAR data that include buildings of all the various roof types to be tested. Therefore, as shown in Figure 17, buildings with various roof types, such as gable, hip, hip and valley, pyramid, gambrel, dormer, sawtooth, dome, and arch, were simulated. The benefit of using simulation data is that the ground truths are known. However, the simulation

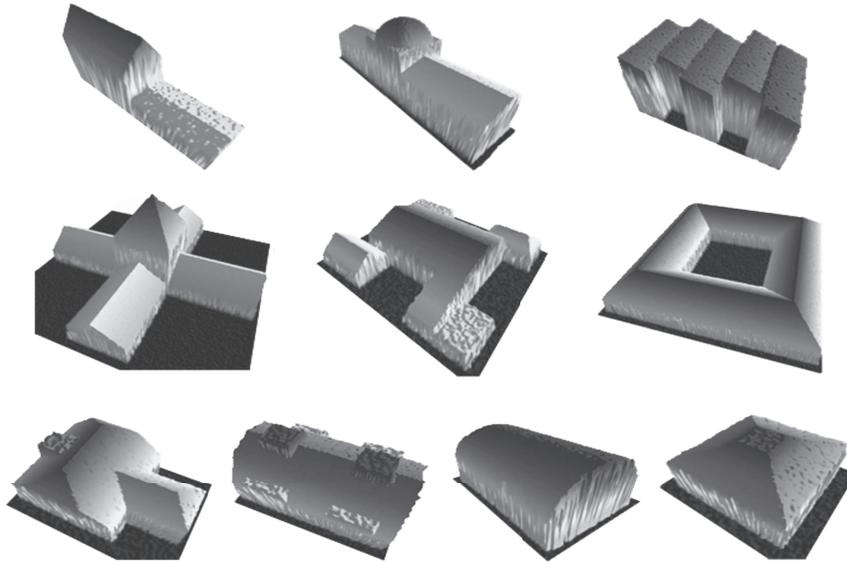


FIGURE 17: Buildings generated from simulated LiDAR data.



FIGURE 18: Vertical components of cube code. (a) Dongtan data. (b) Calgary data.

data do not perfectly reflect the real-world environment. For example, the simulation data do not include trees and vegetation, and the noise characteristics of the simulation data might differ from that of real data.

The average distance between points in the simulation data was 0.25 m, with ± 0.10 m random noise. Real datasets of built-up areas were collected over Dongtan in South Korea and the University of Calgary in Canada, with average point distances of 0.50 m and 0.60 m, respectively. Irregularly distributed LiDAR points were resampled to the regular grid by nearest-neighbor interpolation. The grid cell sizes (i.e., GSD) were equivalent to the average point distances.

3.2. Removal of Trees and Other Vegetation. Figure 18 shows the vertical components of the cube code of the real datasets and Figure 19 shows the building boundaries after removing trees and other vegetation. There was no vegetation in the simulation data. Vertical codes exist not only at the building boundaries but also in the vegetation areas.

However, the cube codes in the latter areas are irregular and could not form closed polygons. Therefore, most of them were removed by the thinning process. However, removing trees and vegetation close to buildings can create boundary shapes somewhat different from the actual shapes. Unwanted objects that remained after this process were removed during segmentation if their size was below a given threshold.

3.3. Results of Segmentation. Figures 20, 21, and 22 present the segmentation and segmented surface boundaries with corner points of the test datasets. With the simulation data (see Figure 20), all the surface patches of the roofs were properly segmented using the multilevel cube. When a fixed uniform cube was used, the surfaces with low slope could not be segmented correctly and could not be distinguished from planar surfaces. However, the curved surfaces (i.e., dome and arch roof) were segmented into several patches because their slope changes from point to point. In this case, the surface type could be recognized by analyzing slope vectors

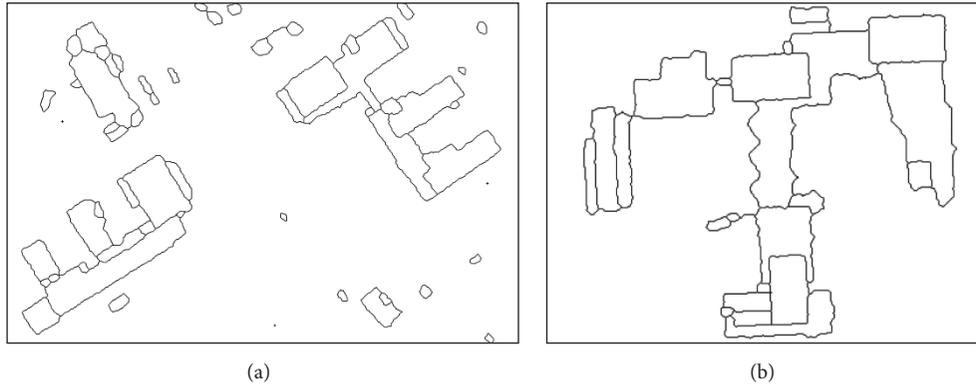


FIGURE 19: Building boundaries after removing trees and vegetation. (a) Dongtan data. (b) Calgary data.

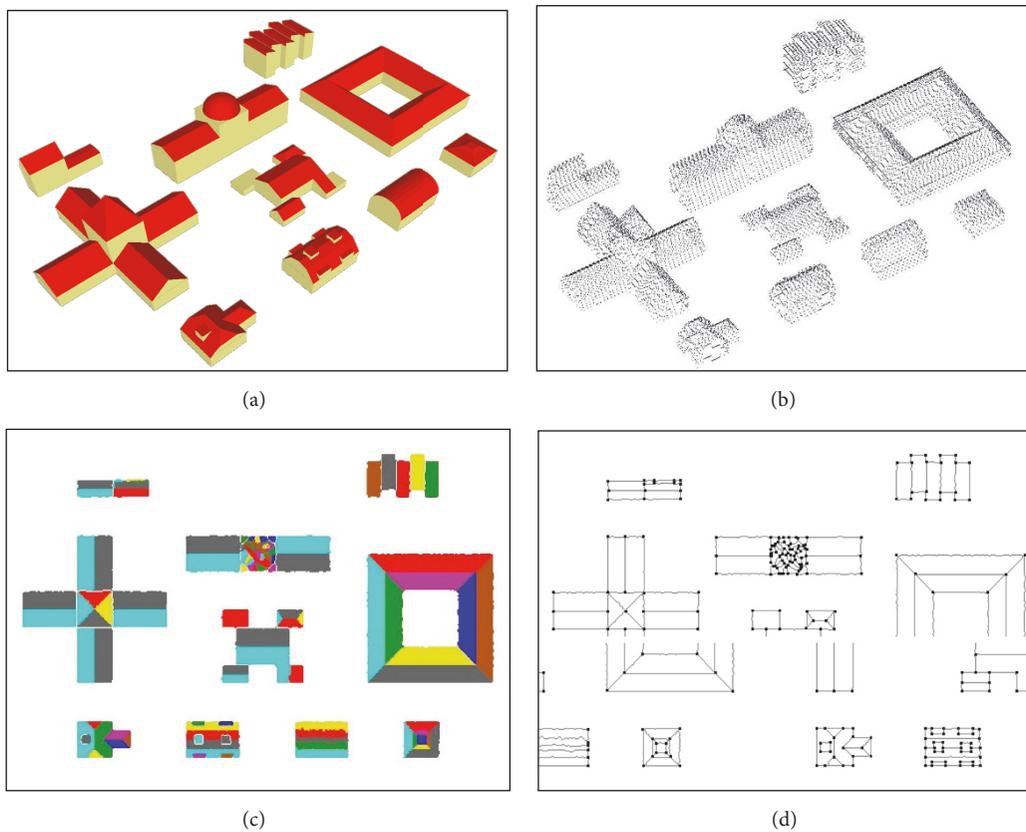


FIGURE 20: Results of simulation data. (a) Building model. (b) LiDAR point clouds. (c) Segmentation. (d) Boundaries of segmented roof and corners.

as described in Section 2.7. In the real datasets (see Figures 21 and 22), most of the roofs have flat surfaces and segmentation was effective, including on gable and arch roofs. The arch roof was segmented into several planar surfaces as with the simulation data.

Most of the corner points of the segmented surfaces were correctly identified. The MKPs of the building boundaries were estimated using these corner points as described in Section 2.6. The MKPs of the building boundaries, determined by the described method, are displayed in the right

column of Figure 23. The left column of Figure 23 shows the corner points of the segmented surface patches. The positional differences between the corner points and the MKPs were computed in terms of root mean-square difference (RMSD).

The RMSDs between the corner points of the segmented surfaces and the MKPs are listed in Table 1. RMSD_X , RMSD_Y , and RMSD_{XY} are the RMSD values of the X - and Y -coordinates and horizontal distance, respectively (i.e., $\text{RMSD}_{XY} = \sqrt{\text{RMSD}_X^2 + \text{RMSD}_Y^2}$). The RMSDs of

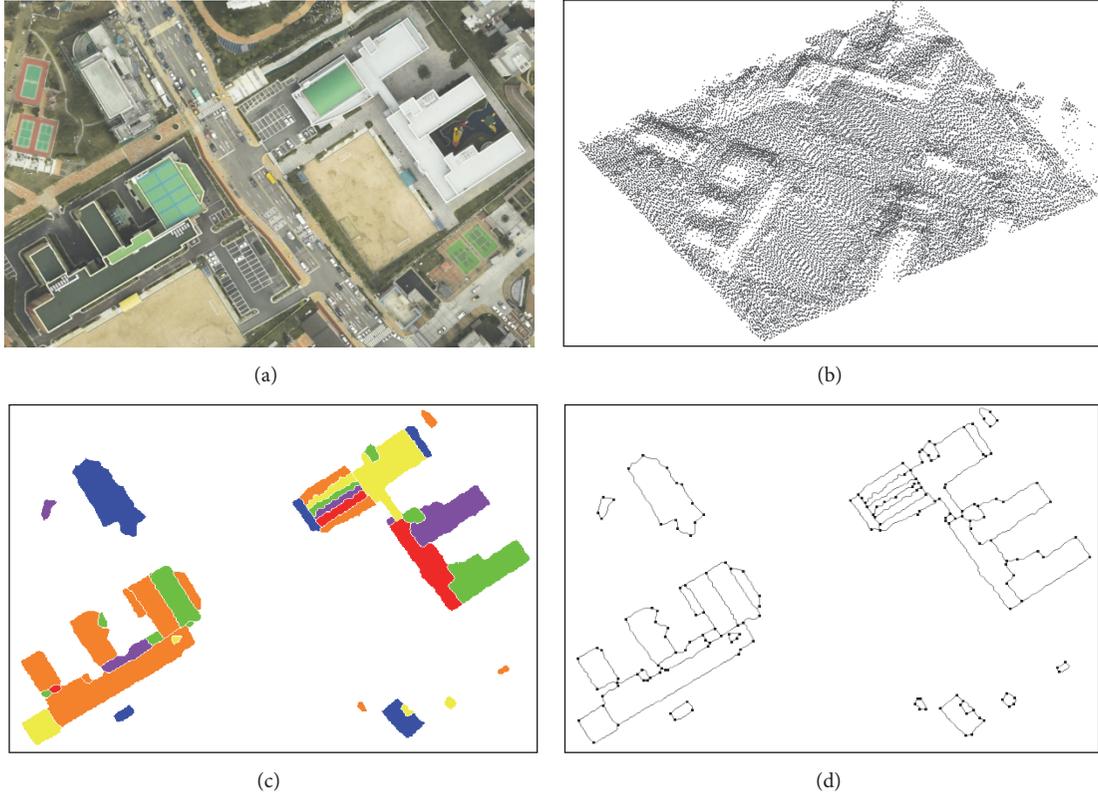


FIGURE 21: Results of the Dongtan data. (a) Building model. (b) LiDAR point clouds. (c) Segmentation. (d) Boundaries of segmented roof and corners.

TABLE 1: RMSDs between corner points of segmented surfaces and MKPs.

Data	GSD[m]	RMSD _X [m]	RMSD _Y [m]	RMSD _{XY} [m]
Simulation	0.25	0.09	0.11	0.14
Dongtan	0.50	0.11	0.18	0.21
Calgary	0.60	0.18	0.19	0.26

the estimated MKPs, compared with the corner points, were nearly all less than half the respective GSDs.

3.4. Results of Curved Surface Modeling. Figures 24, 25, and 26 show the dome and arch roofs of the simulation and real data. The analytical results are presented in Tables 2 and 3. As for the simulation dataset, simulation data 1 is perfect sphere and cylinder with ± 0.10 m random noise, while simulation data 2 is ellipsoidal shape with different point patterns. In Table 2, (X_o, Y_o, Z_o) are the center coordinates and r is the radius of the sphere. In Table 3, (Y_o, Z_o) are the center coordinates and r is the radius of the cylinder. It is noted that the center coordinates (Y_o, Z_o) are the points where the axes intersect the Y - Z plane. α and β are the horizontal and vertical angles of the cylinder axis. $\hat{\sigma}$ values denote a measure of variance. The mean-square error (MSE) and root mean-square error (RMSE) measure the quality of the estimated parameters (i.e., estimators) and the accuracy of the modeling, respectively.

Alternatively, the curved surfaces might be approximated by polynomial functions. Both spherical and cylindrical

surfaces can be modeled by second-order polynomial surfaces through least-square fitting as shown in Figure 27.

$$Z = a_5X^2 + a_4Y^2 + a_3XY + a_2X + a_1Y + a_0 = 0 \quad (8)$$

where (X, Y, Z) are the coordinates of the point clouds and a s are coefficients of the fitting function. Even though the second-order polynomial could not accurately represent the shapes of the surfaces, which are modeled as parabolic surfaces, it might be convenient to use this approach in a situation where high accuracy is not required.

4. Discussion and Conclusions

Segmentation of the LiDAR data, in the form of point clouds, is an essential procedure for most building modeling methods. However, segmentation is a challenging task and a standard method is not yet agreed. Many methods suffer from under- or oversegmentation problems. The proposed method segments roofs of various shapes using cube code that is a 3D extended version of 2D chain code. The key elements for

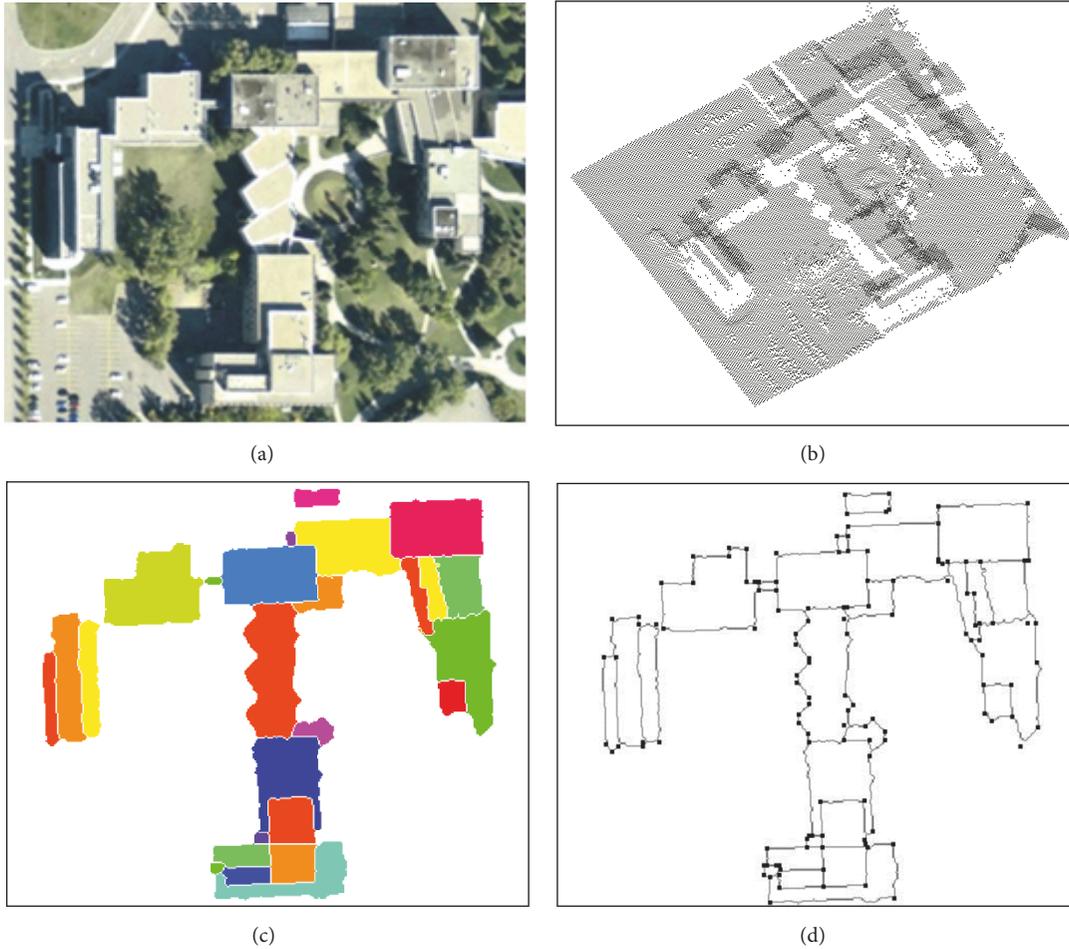


FIGURE 22: Results of the Calgary data. (a) Building model. (b) LiDAR point clouds. (c) Segmentation. (d) Boundaries of segmented roof and corners.

TABLE 2: Parameter estimation and accuracy of spherical surfaces.

Data	Parameter	X_o [m]	Y_o [m]	Z_o [m]	r [m]	$\hat{\sigma}$ [m]	RMSE $_z$ [m]
Simulation 1	Estimation	70.00	65.01	15.00	5.01	0.24	0.06
	MSE of parameter	0.002	0.002	0.003	0.002		
Simulation 2	Estimation	140.01	130.02	5.90	14.31	4.62	0.20
	MSE of parameter	0.015	0.015	0.054	0.047		
Real	Estimation	192411.98	448047.89	79.21	34.06	7.85	0.16
	MSE of parameter	0.001	0.001	0.004	0.003		

shape recognition are the slopes of the surfaces and most shape descriptors utilize slope information. The approach described here segments roof surfaces by encoding surface slopes in 3D space. The described cube operator not only segments roof surfaces but also removes trees and other vegetation, detects building boundaries, and determines MKPs. The accuracy of the MKPs is enhanced by interpolation with adjacent ground points.

The drawbacks of the proposed method are as follows:

- (1) Interpolation of the point clouds is necessary before analysis because of the inherent characteristics of the

cube operator. Interpolation reduces the positional accuracy of the segmentation boundaries and MKPs. Depending on the interpolation method and grid size, the correct breaklines along the boundaries of buildings and roof surfaces might not be wholly preserved.

- (2) Smoothly curved surfaces are segmented into several planar patches (i.e., oversegmentation occurs for such surfaces). However, curved surfaces can be distinguished from planar surfaces by analyzing slope vectors and then modeled with appropriate

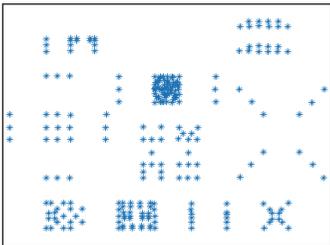
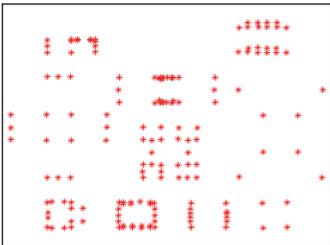
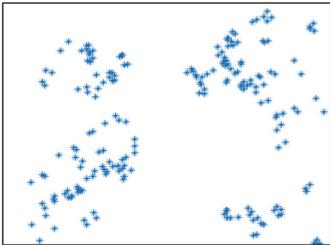
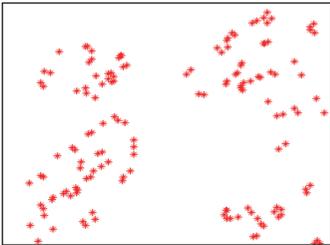
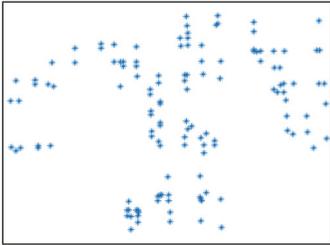
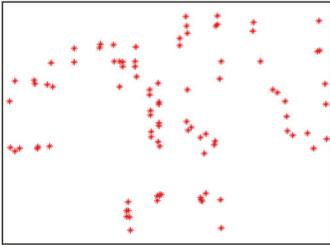
Data	Corner points of segmented surfaces	MKPs of building boundaries
Simulation		
Dongtan		
Calgary		

FIGURE 23: Corner points of segmented surfaces and MKPs of building boundaries.

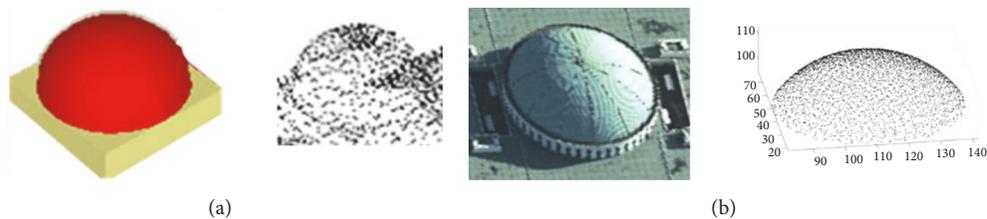


FIGURE 24: Spherical surface. (a) Simulation data 1. (b) Real data.

mathematical functions. The characteristics of the slope vectors were used to identify the type of a curved surface, such as sphere or cylinder.

Based on the described method, the following conclusions were drawn as follows:

- (1) It is possible to segment point clouds by classifying coded surface slope information using a cube operator that is an extension of 2D chain code to 3D.
- (2) A multilevel cube can be adaptively created for identification of more precise slope angles between points. This can overcome the undersegmentation problem; i.e., high LoD segmentation becomes possible.
- (3) Multidirectional processing of the cube operator can solve the rotation variant problem.
- (4) The cube operator can be used to separate ground/nonground features, detect building boundaries and corner points, and remove unwanted features.
- (5) Like 2D chain code, the cube operator provides topological information (i.e., connectivity) between points. Therefore, the boundaries of the buildings can be reconstructed using the corner points of the segmented surfaces.
- (6) MKPs that are close to the real building boundaries can be determined using both the corner points of the segmented surfaces and ground points.

TABLE 3: Parameter estimation and accuracy of cylindrical surfaces. (a) Orientation angles of axis. (b) Centers and radius of cylinder.

(a)					
Data	Parameter	$\alpha[^\circ]$	$\beta[^\circ]$	$\hat{\sigma}_{\alpha 0}[^\circ]$	$\hat{\sigma}_{\beta 0}[^\circ]$
Simulation 1	Estimation	65.23	15.02	1.61	0.15
	MSE of parameter	0.035	0.040		
Simulation 2	Estimation	-29.36	5.00	1.95	0.10
	MSE of parameter	0.010	0.002		
Real	Estimation	33.60	-0.02	2.94	0.13
	MSE of parameter	0.557	0.04		

(b)						
Data	Parameter	$Y_o[m]$	$Z_o[m]$	$r[m]$	$\hat{\sigma}_0[m]$	RMSE $_z[m]$
Simulation 1	Estimation	14.67	4.97	5.00	0.24	0.13
	MSE of parameter	0.001	0.002	0.002		
Simulation 2	Estimation	28.10	-3.42	13.59	6.62	0.32
	MSE of parameter	0.014	0.056	0.049		
Real	Estimation	448047.89	46.91	8.23	3.49	0.23
	MSE of parameter	0.031	0.241	0.228		

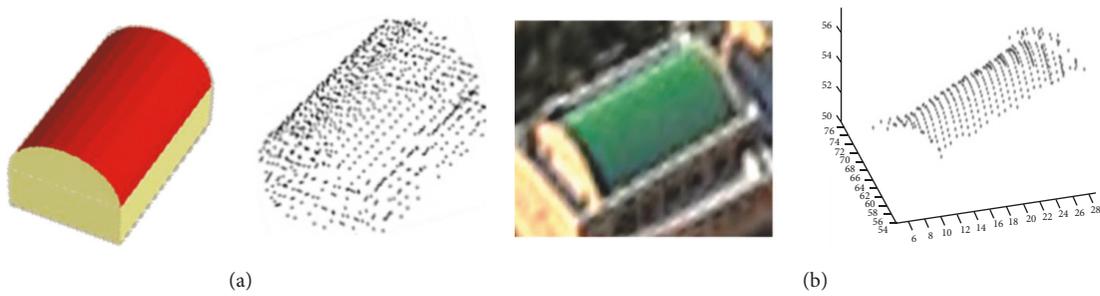


FIGURE 25: Cylindrical surface. (a) Simulation data 1. (b) Real data.

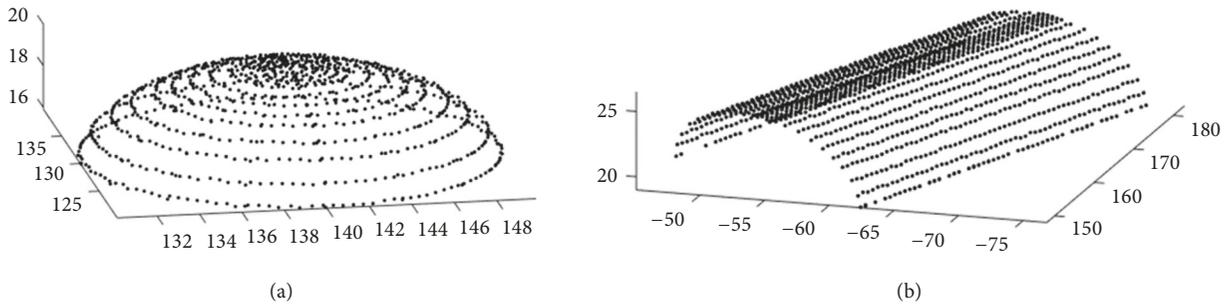


FIGURE 26: Simulation curved surface. (a) Spherical surface 2. (b) Cylindrical surface 2.

Data Availability

The datasets generated during and/or analyzed during the current study are available from the corresponding author on reasonable request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

The authors would like to thank Professor Ayman Habib of the Department of Geomatics Engineering at the University of Calgary for providing airborne LiDAR data. Professor Habib is currently working at Purdue University. This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education [2018R1D1A1B07048732].

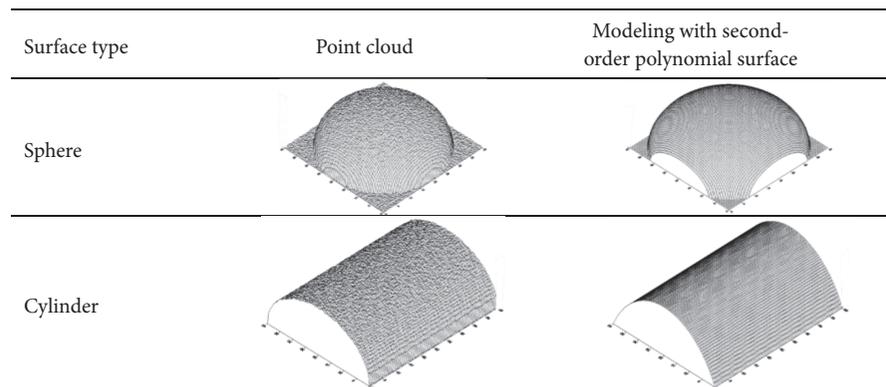


FIGURE 27: Curved surface modeling with polynomial surfaces.

References

- [1] S. A. N. Gilani, M. Awrangjeb, and G. Lu, "Segmentation of airborne point cloud data for building roof extraction," *GIScience and Remote Sensing*, vol. 55, no. 1, pp. 63–89, 2018.
- [2] T. Schenk and B. Csatho, "Fusing Imagery and 3D Point Clouds for Reconstructing Visible Surfaces of Urban Scenes," in *Proceedings of the 2007 Urban Remote Sensing Joint Event*, pp. 1–7, Paris, France, April 2007.
- [3] Z. Lari and A. Habib, "A new approach for segmentation of multi-resolution and multi-platform LiDAR data," in *Proceedings of the Global Geospatial Conference*, Addis Ababa, Ethiopia, November 2013.
- [4] A. Sampath and J. Shan, "Segmentation and reconstruction of polyhedral building roofs from aerial lidar point clouds," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 48, no. 3, pp. 1554–1567, 2010.
- [5] G. Sohn and I. Dowman, "Extraction of buildings from high-resolution satellite data and LIDAR," in *Proceedings of the ISPRS Congress*, Istanbul, Turkey, July 2004.
- [6] D. Zhang and G. Lu, "Review of shape representation and description techniques," *Pattern Recognition*, vol. 37, no. 1, pp. 1–19, 2004.
- [7] J. Zhang, X. Lin, and X. Ning, "SVM-Based classification of segmented airborne LiDAR point clouds in urban areas," *Remote Sensing*, vol. 5, no. 8, pp. 3749–3775, 2013.
- [8] H.-G. Maas and G. Vosselman, "Two algorithms for extracting building models from raw laser altimetry data," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 54, no. 2-3, pp. 153–163, 1999.
- [9] G. Vosselman and S. Dijkman, "3D building model reconstruction from point clouds and ground plans," *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Science*, vol. 34, pp. 37–44, 2001.
- [10] F. Rottensteiner and C. Briese, "Automatic generation of building models from LIDAR data and the integration of aerial images," in *Proceedings of the ISPRS workshop*, vol. 34 of part 3/W13, Dresden, Germany, October 2003.
- [11] G. Miliareisis and N. Kokkas, "Segmentation and object-based classification for the extraction of the building class from LIDAR DEMs," *Computers & Geosciences*, vol. 33, no. 8, pp. 1076–1087, 2007.
- [12] A. Sampath and J. Shan, "Building boundary tracing and regularization from airborne lidar point clouds," *Photogrammetric Engineering and Remote Sensing*, vol. 73, no. 7, pp. 805–812, 2007.
- [13] H. Huang, C. Brenner, and M. Sester, "A generative statistical approach to automatic 3D building roof reconstruction from laser scanning data," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 79, pp. 29–43, 2013.
- [14] D. Chen, R. Wang, and J. Peethambaran, "Topologically aware building rooftop reconstruction from airborne laser scanning point clouds," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 55, no. 12, pp. 7032–7052, 2017.
- [15] E. Loutfia, H. Mahmoud, A. Amr, and S. Mahmoud, "3D model reconstruction from aerial ortho-imagery and LiDAR data," *Journal of Geometry*, vol. 11, no. 1, pp. 38–46, 2017.
- [16] C. Yi, Y. Zhang, Q. Wu et al., "Urban building reconstruction from raw LiDAR point data," *Computer-Aided Design*, vol. 93, pp. 1–14, 2017.
- [17] M. Awrangjeb and C. S. Fraser, "An automatic and threshold-free performance evaluation system for building extraction techniques from airborne LIDAR data," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 7, no. 10, pp. 4184–4198, 2014.
- [18] E. J. Yoo, S. Y. Park, J.-H. Yom, and D.-C. Lee, "Correction of erroneous Model Key Points extracted from segmented Laser Scanner data and accuracy evaluation," *Journal of the Korean Society of Surveying Geodesy Photogrammetry and Cartography*, vol. 31, no. 6, pp. 611–623, 2013.
- [19] H. Freeman, "On the encoding of arbitrary geometric configurations," *IRE Transactions on Electronic Computers*, vol. 2, pp. 260–268, 1961.
- [20] M. Morgan and A. Habib, "Interpolation of LiDAR data and automatic building extraction," in *Proceedings of the ACSM-ASPRS Annual Conference*, Washington, DC, USA, April 2002.
- [21] H.-J. Chu, C.-K. Wang, M.-L. Huang, C.-C. Lee, C.-Y. Liu, and C.-C. Lin, "Effect of point density and interpolation of LiDAR-derived high-resolution DEMs on landscape scarp identification," *GIScience & Remote Sensing*, vol. 51, no. 6, pp. 731–747, 2014.
- [22] Q. Guo, W. Li, H. Yu, and O. Alvarez, "Effects of topographic variability and lidar sampling density on several DEM interpolation methods," *Photogrammetric Engineering and Remote Sensing*, vol. 76, no. 6, pp. 701–712, 2010.

- [23] H. Sánchez-Cruz, H. H. López-Valdez, and F. J. Cuevas, "A new relative chain code in 3D," *Pattern Recognition*, vol. 47, no. 2, pp. 769–788, 2014.
- [24] W. Pratt, *Digital Image Processing*, John Wiley and Sons, New York, NY, USA, 3rd edition, 2001.
- [25] R. Schalkoff, *Digital Image Processing and Computer Vision*, John Wiley and Sons, New York, NY, USA, 1989.

