

Article

Consistent Roof Geometry Encoding for 3D Building Model Retrieval Using Airborne LiDAR Point Clouds

Yi-Chen Chen, Bo-Yi Lin and Chao-Hung Lin * 

Department of Geomatics, National Cheng-Kung University, Tainan 70101, Taiwan;
powcyc@gmail.com (Y.-C.C.); pikadp2004@hotmail.com (B.-Y.L.)

* Correspondence: linhung@mail.ncku.edu.tw; Tel.: +886-6-2757-575

Received: 4 July 2017; Accepted: 17 August 2017; Published: 28 August 2017

Abstract: A 3D building model retrieval method using airborne LiDAR point clouds as input queries is introduced. Based on the concept of data reuse, available building models in the Internet that have geometric shapes similar to a user-specified point cloud query are retrieved and reused for the purpose of data extraction and building modeling. To retrieve models efficiently, point cloud queries and building models are consistently and compactly encoded by the proposed method. The encoding focuses on the geometries of building roofs, which are the most informative part of a building in airborne LiDAR acquisitions. Spatial histograms of geometric features that describe shapes of building roofs are utilized as shape descriptor, which introduces the properties of shape distinguishability, encoding compactness, rotation invariance, and noise insensitivity. These properties facilitate the feasibility of the proposed approaches for efficient and accurate model retrieval. Analyses on LiDAR data and building model databases and the implementation of web-based retrieval system, which is available at <http://pcretrieval.dgl.xyz>, demonstrate the feasibility of the proposed method to retrieve polygon models using point clouds.

Keywords: building modeling; point cloud encoding; spatial histogram; 3D model retrieval

1. Introduction

Recent development on 3D scanning and modeling technologies has led to an increasing number of 3D models, and most of the models have been made available in web-based platforms with data-sharing service. In this context, the question of “How to generate 3D building models?” may evolve to “How to find them in model databases and in the Internet?” [1]. This concept motivates this study to encode unorganized, noisy, and incomplete building point clouds acquired by airborne LiDAR and efficiently retrieve 3D models from databases and the Internet. A set of complete or semi-complete building models is retrieved and reused instead of reconstructing point clouds using complicated modeling techniques [2–5].

The naive approach called text-based retrieval uses keywords in metadata to search for the desired 3D models. This method is simple and efficient, but using keywords as queries suffers from difficulties caused by inappropriate annotations and language varieties. By contrast, content-based retrieval is a promising approach that encodes geometries of queries and models using a shape descriptor and matches the encoded coefficients for data retrieval. Most previous studies on content-based 3D model retrieval take polygon models as input queries [6,7]. These methods can efficiently and accurately extract models from databases. However, obtaining a desired polygon model as input query is difficult, which limits the usage of retrieval systems. With the aid of airborne LiDAR, which has the capability of efficient spatial data acquisitions, a model retrieval by using LiDAR point clouds as input queries is proposed. By consistently encoding point cloud and polygon models, a set of building models similar to an input point cloud in geometry shape can be retrieved for the purposes of data

extraction and efficient cyber city modeling. Based on the concepts of data reuse and crowdsourcing, the proposed system can efficiently construct a 3D city model which is one of key components in virtual geographic environment.

Content-based model retrieval methods can be classified into two categories, *model-based retrieval* and *view-based retrieval*, based on the shape descriptor used in encoding. In model-based retrieval, shape similarities are measured using various geometric descriptors, including shape distribution [6], shape spectral [8], and shape topology [9]. In methods based on shape distribution, geometric features defined over feature spaces are accumulated in bins [6]. A histogram of these bins is utilized as the signature of a 3D model. In shape spectral methods, geometric shapes are transformed to a spectral domain and the spectral coefficients are used in shape matching and retrieval [8]. In topology-based methods, model topologies are represented as skeletons, and retrieval is performed based on the assumption that similar shapes have similar skeletons [9]. Unlike model-based methods, view-based methods represent 3D geometric shapes as a set of 2D projections, and 3D models are matched using their visual similarities instead of geometric similarities [7,10,11]. Each projection is described by image descriptors, and shape matching is reduced to measurement of similarities among the views of the query object and models in the database. Model-based and view-based methods perform well on existing benchmarks for polygon model encoding and retrieval. However, these methods are not designed for unorganized, noisy, sparse, and incomplete point clouds. Recently, Chen et al. [12] proposed point cloud encoding using a set of low-frequency spherical harmonic functions (SHs). With the preprocessing of data resampling and filling, the approach can alleviate the difficulties caused by sparse and incomplete sampling of point clouds. However, the use of low-frequency SHs decreases the ability to distinguish objects with similar geometric shapes thereby leading to ambiguity in shape description.

To improve shape distinguishability, a roof geometry encoding that integrates shape distribution with visual similarity is proposed. The main idea is to represent point clouds and polygon models using top-view depth images that can describe the shapes of building roofs and avoid disturbances from insufficient sampling of building side-views. The depth images are further encoded by geometry features with spatial histograms, which introduce the properties of compact description, rotation invariance, noise insensitivity, and consistent encoding of point clouds and polygon models. These properties lead to a compact storage size and real-time retrieval response time. Furthermore, the visual similarity in depth images and shape distribution in spatial histograms increase the distinguishability of geometric shapes. The remainder of this paper is organized as follows. Section 2 describes the methodology of point cloud encoding and building model retrieval. Section 3 introduces the properties of the proposed encoding method. Section 4 discusses experimental results, and Section 5 presents conclusions and future work.

2. Methodology

2.1. System Overview

Figure 1 schematically illustrates the proposed system which consists of three main steps, depth image generation, data encoding, and data retrieval. In the first step, the building models and point cloud, that is, the input query, are represented by top-view depth images for the geometric description of building roofs. An interpolation process is then applied to the depth image of the point cloud for hole filling to facilitate consistent encoding. In data encoding, a set of geometric features are extracted from the interpolated depth images. The extracted geometric features are encoded by utilizing a spatial histogram with a determined origin, which can provide a rotation-invariance shape description. During data retrieval, the encoded coefficients of input query are matched with those of the models in a database using a shape similarity metric. A set of best-matched models are extracted. In this section, the process of depth image generation is described in Section 2.2, followed by the data encoding and retrieval, which are described in Sections 2.3–2.5.

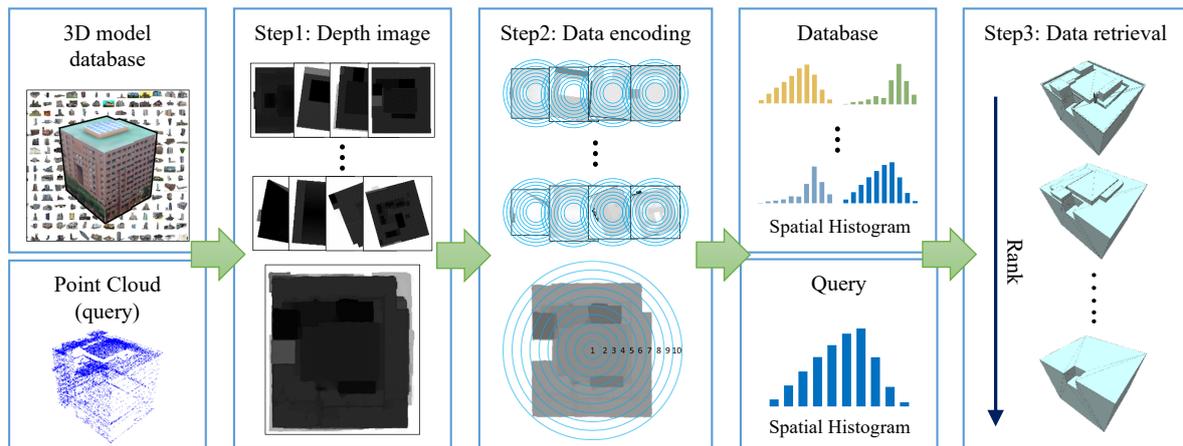


Figure 1. System overview. The system consists of three main steps, depth image generation, data encoding, and data retrieval.

2.2. Generation of Depth Image

Top-view projection is selected in depth image generation because of the characteristics of airborne LiDAR scanning. The roof is generally the most informative and identifiable part of a building in airborne LiDAR acquisitions. In the top-view projection, the pixel value of the projection is generally defined as the height difference between the ground and the pixel in the building roof. However, pixel intensity in this definition is dominated by building height, and the roof geometry is insignificant, especially for tall buildings. In this study, the pixel value of depth image, which is denoted as $D(x, y)$, is defined as the distance between the maximal height of building and the height on that position, that is,

$$D(x, y) = Max_B - H_B(x, y), \quad (1)$$

where Max_B represents the maximal height of a building B , and $H_B(x, y)$ denotes the building height at the position (x, y) . In this definition, the pixel value of depth image corresponds to the relative height difference, which can enhance roof geometry in the depth image.

The depth image generation is a process of rasterization. The spatial resolution of a geometric description depends on the setting of grid size in rasterization. A large grid size indicates low possibility of missing depth information and efficient rasterization, but will result in low spatial resolution of depth images and rough geometric representation of point clouds. By contrast, small grid size is linked to a high-resolution depth image with fine geometric description, but will result in time-consuming computation and large number of empty pixels. In this study, small grid size is preferred because data interpolation can be performed to alleviate the problem of missing information. In addition, the grid size is set according to point cloud density with the expectation that, on average, each grid in the depth image has one inside point. Therefore, grid size is defined as $GS = \sqrt{1/AvgPD}$, where $AvgPD$ represents average point density of a building point cloud. Figure 2 shows an example of point cloud rasterization. The average point density of the building is 16.67 pts/m² and grid size GS is set to 0.245 m in rasterization. In this setting, the geometric detail of building roof is preserved and only small holes are present in the depth image. The search for optimal setting of grid size is difficulty because the determination of grid size is data-sensitive. Any advanced approach on that such as the method proposed by Awrangjeb et al. [13] can be adopted and integrated into the system.

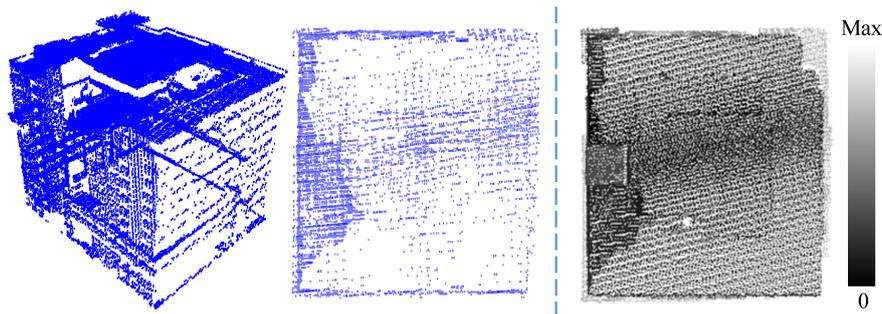


Figure 2. Depth image. (Left, Middle): perspective and top views of point cloud, respectively; (Right): top view of depth image.

Holes are generally present in depth images because of rasterization of irregularly distributed LiDAR point clouds. To address this issue, a hole filling and completion process is performed using grayscale morphological operators. A grayscale morphological closing, which consists of dilation and erosion operators with a flat structuring element, is adopted to fill gaps in depth images. The proposed encoding method requires the preservation of point cloud topology and geometry, but it does not require a perfect depth image, that is, an image without holes. Therefore, a simple and efficient morphology-based approach is adopted. With this hole filling, only holes that are smaller than the structuring element are filled; thus, the geometric topology, such as hollow shape, can be maintained. In the implementation, a 7×7 structuring element is used.

2.3. Data Encoding

The first step of data encoding is determining a translation and rotation invariant origin of a depth image. Chen et al. [12] suggested the selection of the center of minimal bounding box of an object as origin. However, the origin obtained in this approach is slightly sensitive to rotation, especially for non-symmetric buildings. For instance, in Figure 3, the cyan boxes are the minimal bounding boxes of rotated depth images. These bounding boxes differ in height and width, and thus, the obtained origins represented by cyan dots are inconsistent. In this study, a depth image is regarded as a 2.5D model. The barycenter of the 2.5D model is selected as origin because that this center is unchanged after similarity transformation. Examples of obtained origins are shown in Figure 3. The origins displayed in red remain constant as the object is rotated, which implies this origin is invariant to rotation and translation.

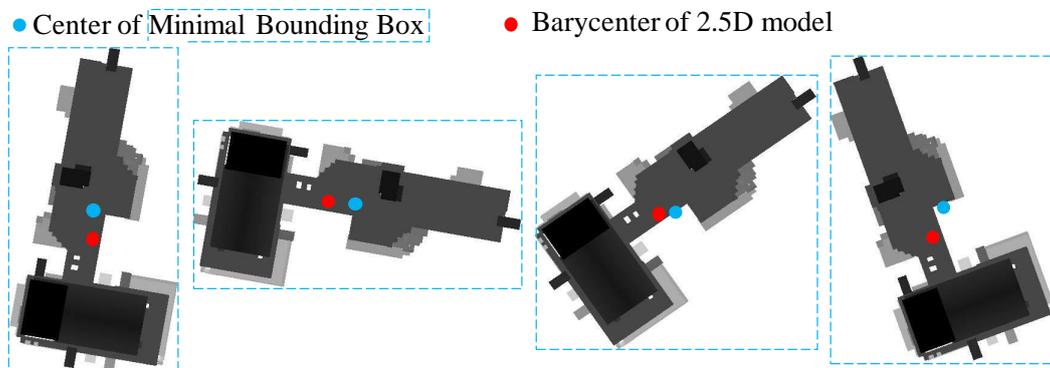


Figure 3. Comparisons between the origins obtained by using the minimal bounding box (the cyan dots) and the building volume (the red dots).

In formula, given a depth image $D(x, y)$, the shape origin (x_0, y_0) is defined as the weighted position in the depth image, that is,

$$(x_0, y_0) = \frac{1}{n} \sum_{i=1}^n (x_i, y_i) \times D_i, \quad (2)$$

where n represents the number of roof pixels, and D_i denotes the value of $D(x_i, y_i)$, which is used as the weight of vector (x_i, y_i) .

Before data encoding, the point cloud and the models are aligned to their origins to facilitate rotation and translation invariant encoding. Three geometric features, namely, height feature, edge feature, and planarity feature, as illustrated in Figure 4, are used to describe the shape of a building roof in depth image domain. These three features are described as follows.

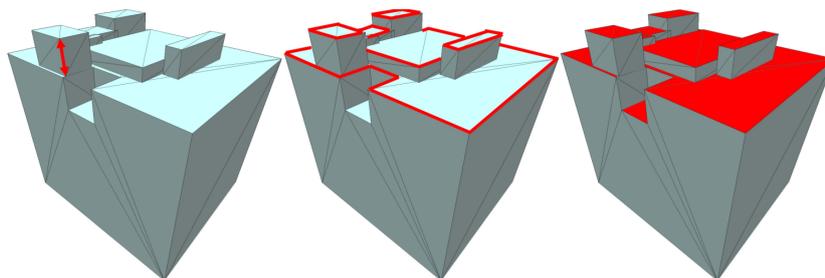


Figure 4. Illustration of roof geometric features. (Left): height feature; (Middle): edge feature; (Right): planarity feature. The features are marked in red.

Height Feature. A building roof is represented as a depth image, and its pixel intensity denotes the relative height of the roof in that position. Therefore, using pixel intensities in a depth image to represent roof geometry is intuitive and efficient. Following the study [14], height feature, which is denoted as $F_h(x, y)$, is defined as the pixel intensities of depth images, that is, $F_h(x, y) = D(x, y)$.

Edge Feature. Sharp edges in an image are linked to high-frequency components that represent details in an image. Therefore, sharp edges are used as geometric features to represent roof geometries. Inspired by the study [15], Laplacian of Gaussian (LoG) filter is adopted to extract sharp edges while alleviating disturbances from noise. The LoG is composed of Gaussian and Laplacian filters, where the low-pass kernel function in the Gaussian filter is used to suppress noise and the second derivative kernel function in Laplacian filter is utilized to extract sharp edges. The LoG is applied to depth images with inherent noises, and the resulting data are used as edge feature, which is denoted by $F_e(x, y)$.

Planarity Feature. Different from sharp edges, which are high-frequency components, planes in an image relate to low-frequency components that represent rough information in an image. Therefore, the planarity feature is selected as geometric feature. The planarity feature from the principal component analysis of a point set is a useful descriptor that can describe the local geometry of a point set and indicate whether local geometry is planar. Given a 2.5D point set $\mathbf{P} = \{(x_i, y_i, D_i)\}_{i=1}^n$ from the pixels in a depth image where the points are located within a circle of diameter r centered at a position $p_c : (x_c, y_c, D_c)$, a simple method for computing the principal components of the point set \mathbf{P} is to diagonalize the covariance matrix of \mathbf{P} . In matrix form, the covariance matrix of \mathbf{P} is written as $\mathbf{C}(\mathbf{P}) = \sum_{p_i \in \mathbf{P}} (p_i - p_c)^T (p_i - p_c)$. The eigenvectors and eigenvalues of the covariance matrix are computed using matrix diagonalization, that is, $\mathbf{V}^{-1} \mathbf{C} \mathbf{V} = \mathbf{D}$, where \mathbf{D} is the diagonal matrix containing the eigenvalues $\{\lambda_1, \lambda_2, \lambda_3\}$ of the covariance matrix \mathbf{C} , and matrix \mathbf{V} contains the corresponding eigenvectors. In geometry, the eigenvalues relate to an ellipsoid that represents the local geometric structure of a point set. The combinations of these eigenvalues provide discriminant geometric features. For instance, $\lambda_1 \cong \lambda_2 \gg \lambda_3$ indicates a flat ellipsoid that can represent planar structures, and $\lambda_1 \cong \lambda_2 \cong \lambda_3$ corresponds to a volumetric structure, such as the corners of buildings.

Following the definitions in [16,17], planarity feature is defined as $(\lambda_2 - \lambda_3)/\lambda_1$, which has the ability of enhancing and describing planar structures. The planarity feature $F_\lambda(x, y)$ obtained from principal component analysis is used as geometric feature.

The height, edge, and planarity features provide the descriptions of height, high-frequency, and low-frequency components of a building roof, respectively. These three features compose a complete set of geometric description for a depth image and a building roof. An example of geometric features is shown in Figure 5. To visualize these features, the feature values are normalized to the range of [0, 255] and displayed as a gray image.

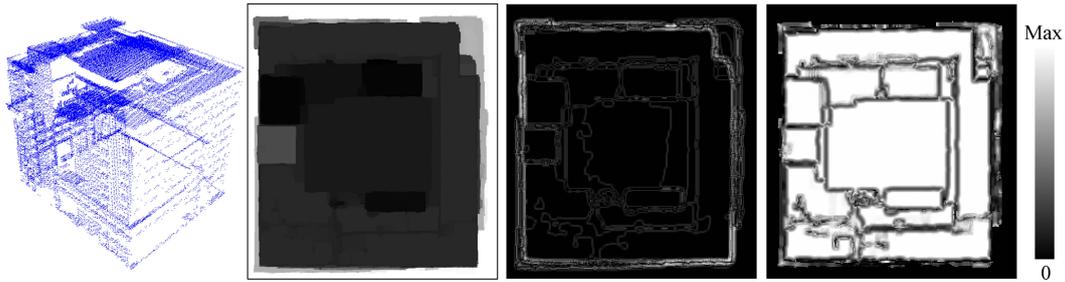


Figure 5. Results of geometric features. (Left): original point cloud; (Right): results of height, edge, and planarity features of the point cloud, respectively.

2.4. Spatial Histogram

The purposes of using spatial histogram in encoding are to reduce the data sizes of geometric features and to achieve rotation-invariance encoding. In spatial histogram generation, the feature image is partitioned into several disjoint circular subspaces or called bins, which are all centered on the determined origin. Similar to image histogram, the intensities of pixels in the feature image are accumulated in bins according to their positions. The accumulated value in each bin is then normalized by the sum of pixel intensities. The range of bin value is [0.0, 1.0] and the sum of all values in bins is 1.0. The distribution of pixel intensities and spatial positions of the pixels are encoded in the histogram, which leads to an encoding with the ability of geometric distinguishability. In the implementation, the feature image is circularly partitioned into k bins with equal width, and the maximum radius of the depth image is set to the distance between the farthest pixel and the determined origin. The number of bins k is set to 30 to balance the encoding compactness and shape distinguishability. A small k may lead to a rough and insufficient description of a geometric shape, whereas a large k may cause inefficient retrieval and increase the sensitivity to noise.

With spatial histogram, the height, edge, and planarity geometric features are described and encoded as (h_1, \dots, h_k) , (e_1, \dots, e_k) , and (p_1, \dots, p_k) , respectively. The total number of components in an encoded coefficient is 90. Examples of spatial histograms with 10 bins are shown in Figure 6. The height, edge, and planarity features used to describe different geometrics have different spatial histogram results.

To further distinguish the building of objects with similar roof geometries but different sizes, the area of the roof, the height of building, and the maximum radius are integrated in the encoding, which can lead to a scale-variant shape representation. The maximal height of a building, denoted as M_h , is defined as

$$M_h = \max(H_B(x, y)) - H_{min}, \quad (3)$$

where the function $\max()$ returns the maximum of an input, and H_{min} represents the minimum of the roof height $H_B(x, y)$. The maximum radius, denoted as M_r , is defined as

$$M_r = \max_radius(\tilde{H}_B(x, y) | H_B(x, y) \notin H_{min}), \quad (4)$$

where the function $max_radius()$ returns the maximal distance in the xy -space between the roof pixel and its origin. The building outliers, that is, $H_B(x, y) = H_{min}$, are excluded from the distance calculation because outliers are generally the protruded exterior facades lying near the roof boundaries. The roof area, denoted by A_f , is defined as

$$A_f = count(\tilde{H}_B(x, y) | H_B(x, y) \notin H_{min}), \quad (5)$$

where the function $count()$ returns the total number of pixels in the input, except pixels with the minimal value H_{min} .

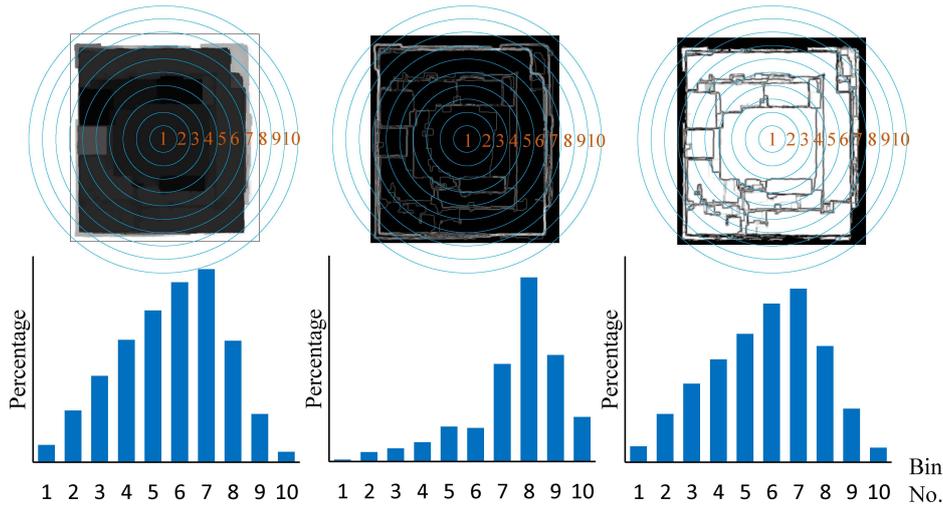


Figure 6. Results of the spatial histograms of height (left); edge (middle); and planarity features (right); Feature images (top) and corresponding spatial histograms of 10 bins (bottom).

2.5. Data Retrieval

By combining the spatial histograms of three geometric features, a point cloud or polygon model is encoded as

$$F_1 = \{(h_1, \dots, h_k), (e_1, \dots, e_k), (p_1, \dots, p_k)\}. \quad (6)$$

In addition, the maximal height, maximal radius, and roof area for scale-variance encoding are combined as

$$F_2 = \{M_h, M_r, A_f\}. \quad (7)$$

Given the encoding in (6) and (7), the measurement of shape similarity is formulated as a combination of F_1 and F_2 . Given a point cloud P and a building model M , shape similarity is defined as

$$dist(P, M) = |F_1(P) - F_2(M)| \times \left(1 + 2 \times \frac{|F_2(P) - F_2(M)|}{|F_2(P) + F_2(M)|} \right), \quad (8)$$

where the first part of the equation, $|F_1(P) - F_2(M)|$, is the measure of geometric similarity between the objects P and M , and the second part, $1 + 2 \times \frac{|F_2(P) - F_2(M)|}{|F_2(P) + F_2(M)|}$, which ranges from 1.0 (no penalty) to 2.0 (maximal penalty), denotes the penalty for different object scales. No penalty is given for buildings with the same scale, and a maximal penalty is set for buildings with extremely large difference on

scale. In (8), a small $dist$ represents a high similarity between the point cloud P and the polygon model M , and vice versa.

The building models downloaded from the Internet are encoded in preprocessing using (6) and (7). The encoded coefficients are stored in a database. When a point cloud is selected as input query in the online retrieval system (<http://pcretrieval.dgl.xyz>), the point cloud is also encoded by using (6) and (7). The obtained coefficients are then matched with the coefficients in the database using (8). Shape similarities are sorted and several top-ranking models are extracted as the query response.

3. Encoding Properties

The proposed encoding method possesses several properties that make a retrieval system feasible to extract polygon building models using point clouds as input queries. The point clouds and polygon models are consistently encoded with the aids of morphological hole-filling and consistent origin determination. To demonstrate this property, a building model and its corresponding point cloud were tested. The encoding results in Figure 7 show that the encoded height, edge, and planarity coefficients of the point cloud and polygon model are similar. This result indicates consistent encoding and the feasibility of retrieving polygon models by using point clouds. Second, the proposed approach provides a shape similarity metric, wherein similar shapes have small distances and dissimilar ones have large distances. This property implies that encoding can distinguish objects with different geometric shapes, which is the foundation of data retrieval.

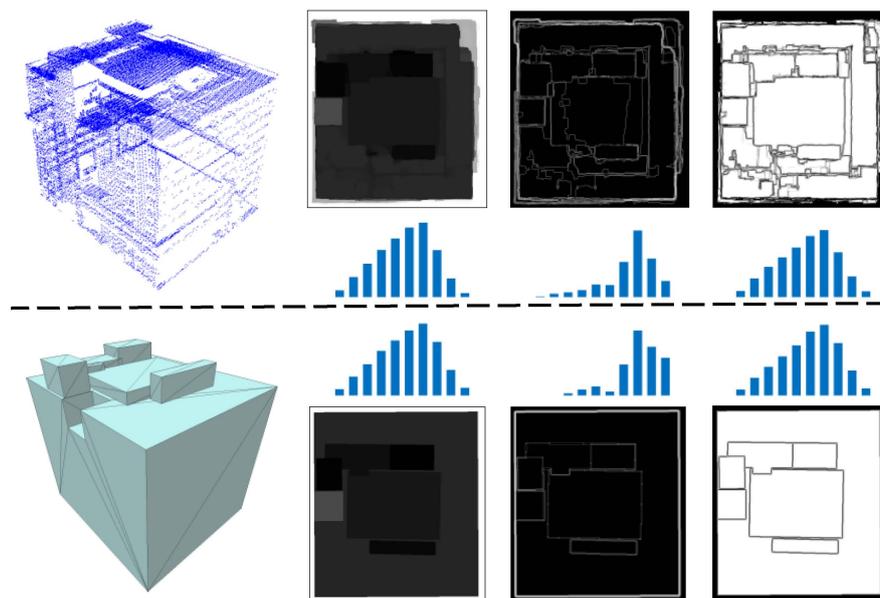


Figure 7. Consistent encoding of point cloud (**top**) and building model (**bottom**). 1st row: original data; 2nd–4th rows: results of height, edge and planarity features and their corresponding spatial histograms.

Third, the encoded coefficients are rotation invariant because of the spatial histogram and the rotation-invariance origin. Figure 8 shows that the coefficients remain unchanged when a 45-degree rotation is applied to the object. Fourth, the proposed encoding scheme is insensitive to noise because the noise problem is alleviated by statistics-based histogram encoding. For example, in Figure 9, a point cloud with Gaussian noise is tested. The standard deviation in the Gaussian noise is set to 0.1. The results show that the encoded coefficients exhibit slight differences when the Gaussian noise is added to the data. Fifth, the building models in the database are compactly encoded. The encoded coefficient for a building model is a set of 90 real numbers, which is smaller than the size of original model data. With the aforementioned properties, the proposed retrieval method can efficiently and accurately retrieve building models using point clouds.

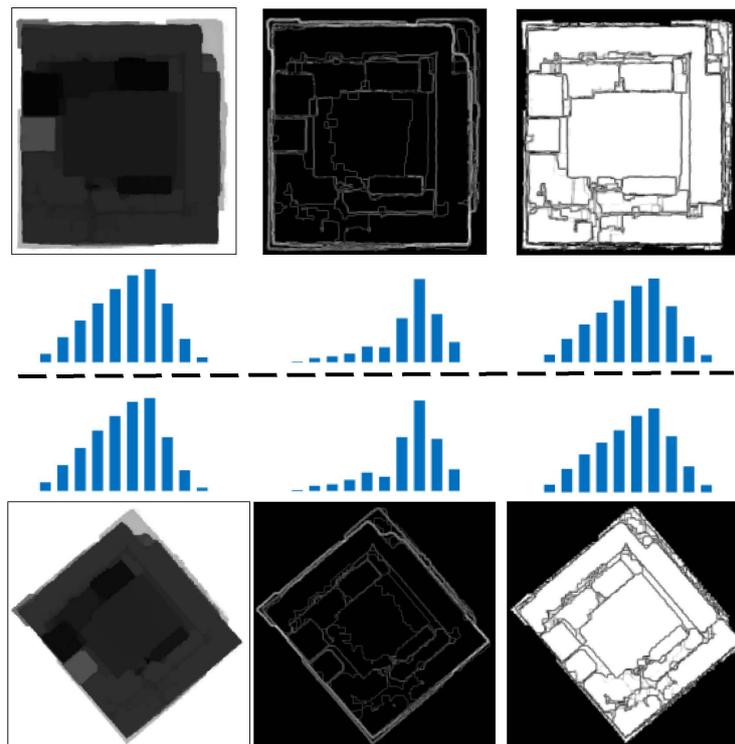


Figure 8. Demonstration of rotation invariance. (**Top**): results of geometric features and corresponding spatial histograms; (**Bottom**): results of rotated data (45-degree rotation).

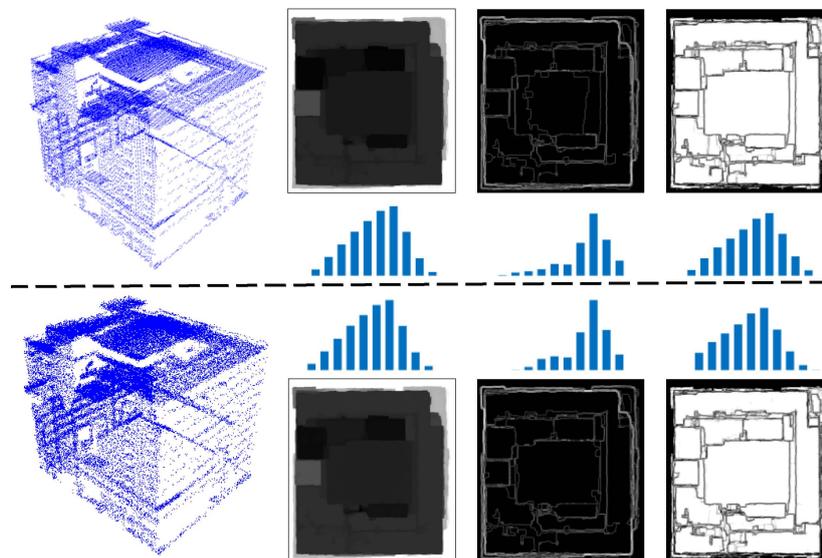


Figure 9. Demonstration of noise insensitivity. (**Top**): point cloud encoding results; (**Bottom**): encoding results of a point cloud with Gaussian noise addition.

4. Experimental Results and Discussion

4.1. Datasets

A database containing about one million 3D building models was tested. The models are downloaded from the Internet using web crawlers that systematically browse the Internet to search for 3D building

models. The study area is the campus of National Cheng-Kung University, Taiwan and Taipei 101 building. The airborne LiDAR point cloud in the study area was acquired by Optech ALTM 30/70 in 2011. This point cloud was extracted from a combination of three overlapped scanning strips. The number of points in the test LiDAR point cloud is 2,772,880 and the average density is 10.72 pts/m² or a point spacing of 0.31 m. The LiDAR point cloud of the study area and its corresponding aerial image are shown in Figure 10. The trees nearby buildings were removed and building point clouds were segmented and extracted from the test data manually for simplicity. Any advanced point cloud segmentation or building extraction algorithms [18–20] can be adopted to extract building point clouds automatically or semi-automatically.

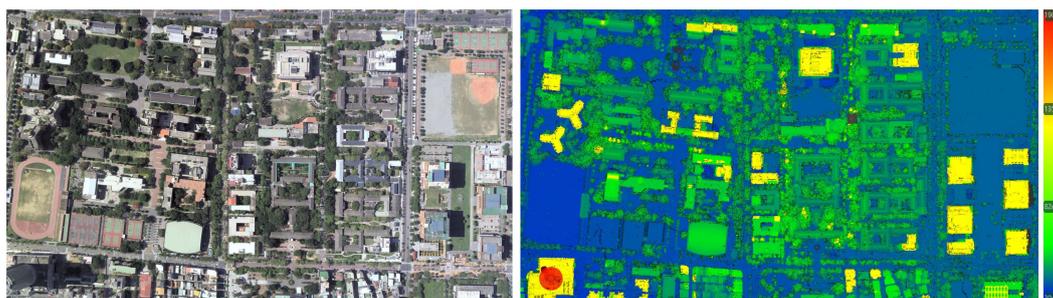


Figure 10. Study area. The aerial image (left) and corresponding point cloud (right).

4.2. Computational Performance

The encoding time for a point cloud of 22,440 points and about 2000 m² areas is 150 ms, and the retrieval response time is 600 ms for a database containing one million models. The information of the tested building point clouds, grid sizes, and computation time are shown in Table 1. The grid sizes are automatically calculated according to the average point density of the building. Encoding time mainly depends on the number of points and the xy-area of a building, and the retrieval time is linearly dependent on the size of the model database.

Table 1. Time performance. 1st column: point clouds of the tested building; 2nd column: number of points; 3rd column: ground area of a building; 4th column: average point density; 5th column: grid size; 6th column: encoding time (En.); 7th column: retrieval time (Re.).

Point Clouds	#Points	Area (m ²)	Avg.PD	GS (m)	Time Performance	
					T_en(ms)	T_re(ms)
	50843	3050.1	16.67	0.245	155	606
	183719	9848.6	18.65	0.232	413	597
	26077	2141.750	12.18	0.287	155	579
	55061	3829.500	14.38	0.264	205	637
	29698	1342.438	22.12	0.212	155	606

4.3. Evaluation of Model Encoding

To evaluate the proposed method, two sets of building models that have different level-of-details (LODs) on roofs were tested. The purpose of this experiment is to evaluate the distinguishability of encoding approaches on building models with different geometric roof details. The first dataset that contains four models is a building with hollow geometry. The original model with hollow geometry, which is denoted by LoD_1^a , is gradually reduced to a non-hollow model with simplified roof geometry, which is denoted as LoD_4^a . The simplified and detached parts of the buildings are marked by red. The second dataset contains five models, where the original model with detailed geometry on roof, denoted by LoD_1^b , is gradually simplified to be a simple box, denoted by LoD_5^b . The aerial photos, LiDAR point clouds, and well-constructed building models, which are, LoD_1^a and LoD_1^b , are shown in Figure 11. The point clouds and the models are encoded using the proposed method. The shape similarities between the models and their corresponding point clouds are measured using (8). The models and their similarity values are shown in Figure 12. For the first dataset, the results indicate that the proposed encoding method can separate the model with hollow geometry from the model without hollow geometry. The results in the second dataset show that the shape similarities decrease near-linearly when the geometry of the original model is simplified gradually. These experiments demonstrates the ability to describe hollow geometric shapes and distinguish models with different details on roof geometry, which are superior over the related method [12] that encodes models using a set of low-frequency spherical harmonic functions. Because of the use of low-frequency and spherical basis functions, the method [12] is not able to distinguish models with similar geometry details and models with and without hollow geometries.

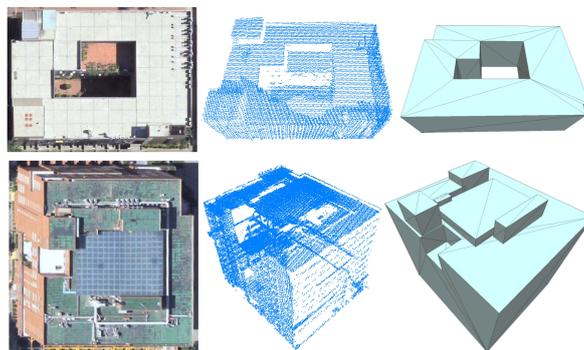


Figure 11. Tested building models. (Left): orthogonal image; (Middle): airborne LiDAR point clouds; (Right): 3D models downloaded from the Internet.

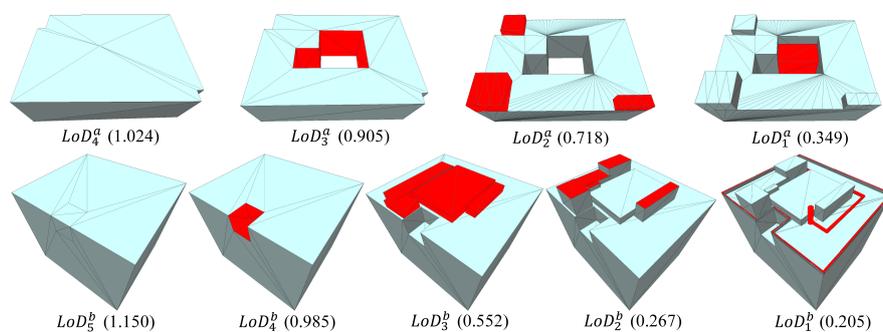


Figure 12. Tested models with different LODs. The numbers shown below the models represent the shape similarities between the models and their corresponding point clouds.

4.4. Evaluation of Model Retrieval

The proposed method was compared with the method [12] by using a database that contains around one million 3D building models downloaded from the Internet. Three building point clouds were tested as the input queries. To evaluate the retrieval accuracy, the commonly used measurement, namely, root mean square deviation (RMSD), is adopted to calculate the shape similarities between the input query and the retrieved models. Before RMSD calculation, the retrieved models are aligned with the input query semi-automatically using the standard registration algorithm called iterative closet point [21]. The shape similarities obtained by RMSD are used as reference values in retrieval evaluation. Figure 13 shows the retrieval results, including the retrieved models, ranks, and reference RMSD values. The visual comparisons show that the models extracted by these two methods are similar to the queries in shape, and the ranking generated by the proposed method is better than that in [12], especially for the third data. This experiment shows that the proposed method combining shape distribution with visual similarity can improve the ability to distinguish geometric shapes, compared with [12]. However, the ranks obtained by these two approaches are not well-matched with that of RMSD references. For instance, in the first test data, the RMSD of the ranked 5th model is larger than that of rank 7th model, which means that the ranking is inconsistent with the reference. The imperfect ranking is caused by the use of compact shape description. To achieve efficient retrieval, data are compactly encoded by shape description, and only 90 coefficients are used in the proposed method to encode a point cloud or a polygon model. To solve this problem, the ranks of the extracted models can be further refined using RMSD measurement if perfect ranking is required.

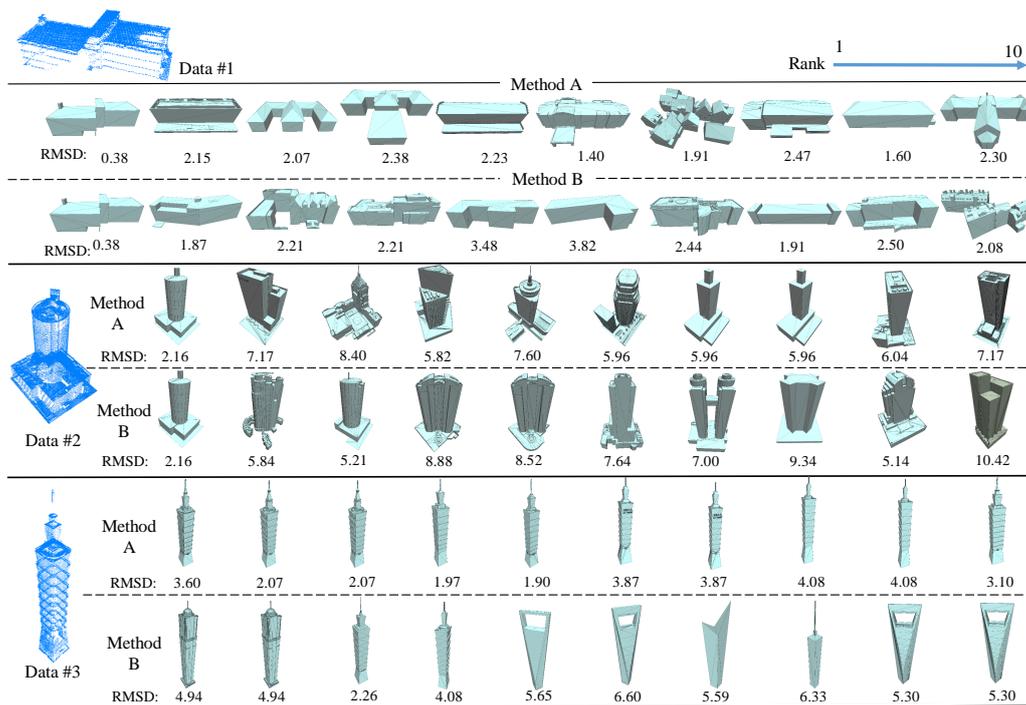


Figure 13. Comparisons among the retrieval results obtained by our method (Method A) and the related method [12] (Method B).

To further analyze retrieval performance, the ranking of all extracted models by using RMSD is used as reference. Then, the measurement for ranking is defined as the difference between the obtained ranking and reference, that is, $Diff_r = |R_{method} - R_{ref}|$, where R_{ref} and R_{method} represent the reference ranking and the ranking from a method, respectively. In this measure, the average of $Diff_r$, which is denoted as $Avg.Diff_r$, is used to estimate ranking quality. The statistical analysis for $Avg.Diff_r$ is

shown in Table 2. A small $Avg.Diff_r$ means high-quality ranking. In addition, the commonly used measurements precision η_s and recall η_n were adopted to evaluate the retrieval accuracy using the Data #3 in Figure 13. These measurements are defined as $\eta_s = \frac{TP}{TP+FP}$ and $\eta_n = \frac{TP}{TP+FN}$, where TP , FP , and FN represent true positive, false positive, and false negative, respectively. The results are shown in Table 3. From the statistical analysis, the retrieval results by the proposed method are superior to that by [12].

Table 2. Comparisons between retrieved rankings using the proposed method and the method by Chen et al. (2014). 1st column: RMSD of the retrieved model; 2nd column: reference ranking; 3rd column: $Diff_r$ value. Two numbers in a pair denotes the performances of the compared methods.

The Proposed Method/ Chen et al. (2014)									
Rank	Data #1			Data #2			Data #3		
	RMSD	Ref.	$Diff_r$	RMSD	Ref.	$Diff_r$	RMSD	Ref.	$Diff_r$
1	0.38/0.38	1/1	0/0	2.16/2.16	1/1	0/0	3.60/4.95	7/13	6/12
2	2.15/1.87	9/4	7/2	7.18/5.84	12/5	10/3	2.07/4.95	3/12	11/0
3	2.07/2.21	7/10	4/7	8.40/5.21	15/3	12/0	2.07/2.26	3/5	0/2
4	2.38/2.21	14/11	10/7	5.83/8.88	4/17	0/13	1.97/4.08	2/10	2/6
5	2.23/3.48	12/18	7/13	7.60/8.52	13/16	8/11	1.90/5.65	1/17	4/12
6	1.40/3.82	2/19	4/13	5.96/7.64	6/14	0/8	3.87/6.60	9/19	3/13
7	1.91/2.44	6/15	1/8	5.96/7.00	7/10	0/3	3.87/5.59	8/16	1/9
8	2.47/1.91	16/5	8/3	5.96/9.34	7/18	1/10	4.08/6.33	10/18	2/10
9	1.59/2.50	3/17	6/8	6.04/5.14	9/2	0/7	4.08/5.30	10/14	1/5
10	2.30/2.08	13/8	3/2	7.17/10.42	11/19	1/9	3.10/5.30	6/14	4/5
Avg.$Diff_r$	1.89/2.30			6.23/7.01			3.06/5.10		

Table 3. Precision-and-recall of our method (Method A) and the related method [12] (Method B). The tested data is Data #3 in Figure 13 is used as tested data.

	# of queries	1	2	3	4	5	6	7	8	9	10	11	12	13	14
MethodA	Precision	0.00	0.00	0.33	0.50	0.40	0.33	0.29	0.25	0.22	0.20	0.18	0.25	0.23	0.21
	Recall	0.00	0.00	0.07	0.14	0.14	0.14	0.14	0.14	0.14	0.14	0.14	0.21	0.21	0.21
MethodB	Precision	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.93
	Recall	0.07	0.14	0.21	0.29	0.36	0.43	0.50	0.57	0.64	0.71	0.79	0.86	0.93	0.93

5. Conclusions and Future Work

This study proposed a new method for 3D building model retrieval using LiDAR point clouds as input query. To archive consistent encoding of polygonal models and point clouds, rotation-invariance origin determination is adopted, which utilizes object volume to determine the origin instead of using the minimal bounding box of an object. In addition, the morphological closing operator is applied to fill the holes in the depth images of point clouds. Filling holes effectively alleviates the difficulties caused by sparse and incomplete sampling of point clouds and facilitates consistent encoding. The proposed encoding that integrates shape distribution with visual similarity can increase the distinguishability of geometric shapes. The experiments on building models with different LoDs show that ambiguous shape description is avoided. The experiments also demonstrate that using the spatial histogram of geometric features as shape descriptor introduces the properties of noise-insensitivity and rotation-invariance to the retrieval. These properties facilitate the feasibility of the proposed approaches to achieve efficient and accurate model extraction. Based on the qualitative and quantitative analyses on LiDAR data and the building models and based on the implemented web-based retrieval system, we conclude that retrieve building models by using point clouds is feasible. In the future, we plan to extend the proposed method to point clouds for photogrammetry techniques and terrestrial LiDAR, which also have a great need of 3D modeling.

Acknowledgments: The authors would like to thank the anonymous reviewers for their insightful comments. This work was supported in part by the National Science Council (Contracts MOST 104-2628-M-006-002-MY3 and MOST 104-2628-E-006-003-MY3), Taiwan.

Author Contributions: Yi-Chen Chen conducted the research, performed the experiments and implemented the retrieval system; Bo-Yi Lin designed the website interface of retrieval system; Chao-Hung Lin contributed to the analysis and to writing.

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

References

1. Funkhouser, T.; Min, P.; Kazhdan, M.; Chen, J.; Halderman, A.; Dobkin, D.; Jacobs, D. A search engine for models. *ACM Trans. Graph.* **2003**, *22*, 83–105.
2. Huang, H.; Brenner, C.; Sester, M. A generative statistical approach to automatic 3d building roof reconstruction from laser scanning data. *ISPRS J. Photogramm. Remote Sens.* **2013**, *79*, 29–43.
3. Jarzgbek-Rychard, M.; Borkowski, A. 3D building reconstruction from ALS data using unambiguous decomposition into elementary structures. *ISPRS J. Photogramm. Remote Sens.* **2016**, *118*, 1–12.
4. Perera, G.S.N.; Maas, H.-G. Cycle graph analysis for 3D roof structure modelling: concepts and performance. *ISPRS J. Photogramm. Remote Sens.* **2014**, *93*, 213–216.
5. Xiong, B.; Elberink, S.O.; Vosselman, G. A graph edit dictionary for correcting errors in roof topology graphs reconstructed from point clouds. *ISPRS J. Photogramm. Remote Sens.* **2014**, *93*, 227–242.
6. Akgul, C.B.; Sankur, B.; Yemez, Y.; Schmitt, F. 3D model retrieval using probability density-based shape descriptors. *IEEE Trans. Pattern Anal. Mach. Intell.* **2009**, *31*, 1117–1133.
7. Gao, Y.; Wang, M.; Zha, Z.-J.; Tian, Q.; Dai, Q.-H.; Zhang, N.-Y. Less is more: efficient 3D object retrieval with query view selection. *IEEE Trans. Multimedia* **2011**, *13*, 1007–1018.
8. Mademlis, A.; Daras, P.; Tzovaras, D.; Strintzis, M.G. Ellipsoidal harmonics for 3-D shape description and retrieval. *IEEE Trans. Multimedia* **2009**, *11*, 1422–1433.
9. Tam, K.L.; Lau, W.H. Deformable model retrieval based on topological and geometric signatures. *IEEE Trans. Vis. Comput. Graph.* **2007**, *13*, 470–482.
10. Chen, D.Y.; Tian, X.P.; Shen, Y.T. On visual similarity based 3D model retrieval. *Comput. Graph. Forum* **2003**, *22*, 223–232.
11. Papadakis, P.; Pratikakis, I.; Perantonis, S.; Theoharis, T. Efficient 3D shape matching and retrieval using a concrete radialized spherical projection representation. *Pattern Recognit.* **2007**, *40*, 2437–2452.
12. Chen, J.-Y.; Lin, C.-H.; Hsu, P.-C.; Chen, C.-H. Point cloud encoding for 3D building model retrieval. *IEEE Trans. Multimedia* **2014**, *16*, 337–345.
13. Mohammad, A.; Zhang, C.; Fraser, C.S. Building Detection in Complex Scenes Thorough Effective Separation of Buildings from Trees. *Photogramm. Eng. Remote Sens.* **2012**, *78*, 729–745.
14. Mallet, C.; Bretar, F.; Roux, M.; Soergel, U.; Heipke, C. Relevance assessment of full-waveform lidar data for urban area classification. *ISPRS J. Photogramm. Remote Sens.* **2011**, *66*, s71–s84.
15. Maltezos, E.; Ioannidis, C. Automatic Detection of Building Points from LIDAR and Dense Image Matching Point Clouds. *ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci.* **2015**, *II-3/W5*, 33–40.
16. Gross, H.; Thoennessen, U. Extraction of lines from laser point clouds. *Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci.* **2016**, *36*, 86–91.
17. Lin, C.-H.; Chen, J.-Y.; Su, P.-L.; Chen, C.-H. Eigen-feature analysis of weighted covariance matrices for LiDAR point cloud classification. *ISPRS J. Photogramm. Remote Sens.* **2014**, *94*, 70–79.
18. Li, Y.; Hu, Q.; Wu, M.; Liu, J.; Wu, X. Extraction and Simplification of Building Facade Pieces from Mobile Laser Scanner Point Clouds for 3D Street View Services. *ISPRS Int. J. Geoinf.* **2016**, *5*, 231.
19. Serna, A.; Marcotegui, B.; Hernandez, J. Segmentation of Facades from Urban 3D Point Clouds Using Geometrical and Morphological Attribute-Based Operators. *ISPRS Int. J. Geoinf.* **2016**, *5*, 6.
20. Mongus, D.; Lukač, N.; Žalik, B. Ground and building extraction from LiDAR data based on differential morphological profiles and locally fitted surfaces. *ISPRS J. Photogramm. Remote Sens.* **2014**, *93*, 145–156.
21. Besl, P.J.; McKay, N.D. Method for registration of 3D shapes. *IEEE Trans. Pattern Anal. Mach. Intell.* **1992**, *14*, 239–256.

