

Point Cloud Encoding for 3D Building Model Retrieval

Jyun-Yuan Chen, Chao-Hung Lin, Po-Chi Hsu, Chung-Hao Chen

Abstract—An increasing number of three-dimensional (3D) building models are being made available on Web-based model-sharing platforms. Motivated by the concept of data reuse, an encoding approach is proposed for 3D building model retrieval using point clouds acquired by airborne light detection and ranging (LiDAR) systems. To encode LiDAR point clouds with sparse, noisy, and incomplete sampling, we introduce a novel encoding scheme based on a set of low-frequency spherical harmonic basis functions. These functions provide compact representation and ease the encoding difficulty coming from inherent noises of point clouds. Additionally, a data filling and resampling technique is proposed to solve the aliasing problem caused by the sparse and incomplete sampling of point clouds. Qualitative and quantitative analyses of LiDAR data show a clear superiority of the proposed method over related methods. A cyber campus generated by retrieving 3D building models with airborne LiDAR point clouds demonstrates the feasibility of the proposed method.

Index Terms—point cloud encoding, 3D model retrieval, cyber city modeling.



1 INTRODUCTION

The problem of 3D building model retrieval using airborne LiDAR point clouds as input queries is addressed in this paper. LiDAR is an optical scanning technique that is capable of measuring the distance to a target object. By integrating global positioning system (GPS) and inertial navigation system (INS), an airborne LiDAR gains the capability to acquire high-resolution point clouds from objects in the ground efficiently and accurately. Thus, an airborne LiDAR system can provide surveyors with the capability of digital elevation and cyber city model generation [1]. This study aims at the efficient construction of a cyber city by encoding unorganized, noisy, and incomplete point clouds, as well as by retrieving 3D building models from model databases or from the Internet.

Recent developments on Web 2.0 technique and scanning equipment have yielded an increasing number of 3D building models in Web-based data-sharing platforms. Based on the concept of data reuse, a complete or semi-complete building model in databases or in the Internet is reused rather than reconstructing point cloud. The main theme of model retrieval is accurate and efficient representation of a 3D shape. Existing studies mainly focus on encoding and retrieving 3D polygon models using polygon models as input queries [2–7]. These studies do not consider model retrieval by using point clouds, which is in a great need in the topic of efficient cyber city construction with LiDAR point clouds.

The key idea behind the proposed method is to represent noisy point clouds using a complete set of spherical harmonics (SHs). Point clouds represented by a few low-frequency SHs are insensitive to noises. In addition,

SH encoding reduces data description dimensions and yields a compact shape descriptor, resulting in both storage size and search time reduction. Moreover, the inherent rotation-invariant property and multi-resolution nature of SH encoding enable the efficient matching and indexing of the model database. Besides, a data filling and resampling approach is proposed to solve encoding problems coming from the incomplete shapes of point clouds and the aliasing problems of SH coefficients attributed to the sparse sampling of point clouds.

Although the use of SHs in general data retrieval has previously been studied [2, 8], to the best of our knowledge, the proposed method is the pioneer in utilizing SHs to encode airborne LiDAR point clouds, i.e., unorganized, noisy, sparse, and incomplete data. The proposed method, therefore, presents two major contributions: 1) the introduction of an encoding scheme that consistently encodes input point clouds and 3D building models in the database, and 2) the application of the proposed encoding scheme to retrieve 3D building models for efficient cyber city construction. The remainder of this paper is organized as follows. Section 2 reviews related works. Section 3 describes the methodology of point cloud encoding and building model retrieval. Section 4 discusses the experimental results, and section 5 presents the conclusions and plans for future work.

2 RELATED WORK

Only studies that are closely related to the proposed work are presented in this section. For detailed surveys on shape retrieval, please refer to [9]. Following the categorization presented by Akgul et al. [6], 3D model retrieval methods are classified into two categories: retrieval based on *projected views* and *3D descriptors*. For retrieval methods based on projected views, 3D shapes

are represented as a set of 2D projections [5, 7, 10–12]. Each projection is described by image descriptors. Thus, shape matching is reduced to measure similarities between views of the query object and those of the models in the database. Although methods based on projected views can yield good retrieval results, a large number of views may degrade retrieval efficiency. In addition, extracting accurate and complete shapes from the projected views of unorganized and incomplete point clouds is difficult.

For retrieval methods based on 3D descriptors, shape similarities are measured by using various geometric shape descriptors including shape topology [13–17], shape distribution [3, 6, 18], SH [2, 4, 8], shape spectral [19], Zernike moment [20], angular radial transform [21], and radon transform [22]. In the topology-based methods [13–17], shape topologies are generally represented as skeletons/graphs. These methods rely on the fact that the skeleton is a compact shape descriptor, and assume that similar shapes have similar skeletons. These conditions enable a topology-based method to facilitate efficient shape matching and even partial shape matching [16]. In the methods of shape distribution [3, 6], geometric features are accumulated in bins that are defined over feature spaces. A histogram of these values is then used as the signature of a 3D model. In the transformation-based methods [2, 4, 8, 19–22], 3D shapes are transformed to other domains, and transformation coefficients are used in shape matching and retrieval. Among them, SH is the typical transformation [2, 4, 8]. By utilizing the advantages of compact shape description and rotation invariant, models can be efficiently retrieved. Similarly, the spectral embedding proposed by Jain and Zhang [19] represents 3D models using eigenvectors of the matrix formed by shape spectral. Thus, this descriptor can withstand the surface disturbance of articulated shape deformation in addition to similarity transformation. The 3D Zernike moment proposed by Novotni and Klein [20], the angular radial transform proposed by Ricard et al. [21], and the radon transform proposed by Daras et al. [22] are other kinds of transformation-based approaches.

The use of these 3D descriptors in shape retrieval is not a novel concept. However, substantial differences exist between our method and the related methods. The main difference is that the related methods perform very well on existing benchmarks [23] for encoding and retrieving 3D polygon models. However, these methods cannot be applied to unorganized, noisy, sparse, and incomplete 3D point clouds, which is in a great need in the topic of efficient city model construction. Consistently and accurately encoding LiDAR point clouds and building models is the goal of this study.

3 METHODOLOGY

3.1 System Overview

Fig. 1 schematically illustrates the workflow of the proposed system, which comprises two major components:

data encoding and *data retrieval*. For data encoding, both the building models and point clouds (i.e., the input query) are consistently encoded by a set of SHs. To achieve consistency in encoding, an airborne LiDAR simulator is utilized to resample the building models. This process enables building models and point clouds to have similar samplings, as shown in Fig. 1(b). In addition, a data filling process is performed to solve the aliasing problem attributed to the sparse sampling of point clouds, as shown in Figs. 1(c) and 1(f). This process can significantly reduce aliasing errors on encoded SH coefficients, thus improving retrieval accuracy. For data retrieval, a LiDAR point cloud is selected as a query input to retrieve building models in the database. The data are retrieved by matching the SH coefficients of point clouds and building models. By utilizing the inherent multi-resolution property of SH representation, a coarse-to-fine efficient indexing and ranking scheme is adopted to accelerate model retrieval.

3.2 Spherical Harmonics

To represent point clouds and building models efficiently, a set of SH functions are used for geometric shape encoding. The following is a brief introduction to SHs. A SH function of degree ℓ and order m , denoted by $Y_\ell^m(\theta, \phi)$, is defined as follows:

$$Y_\ell^m(\theta, \phi) = \sqrt{\frac{2\ell + 1}{4\pi} \frac{(\ell - m)!}{(\ell + m)!}} P_\ell^m(\cos\theta) e^{im\phi}, \quad (1)$$

where ℓ and m are integers that satisfy $\ell \geq 0$ and $|m| \leq \ell$; $\theta \in [0, \pi]$ and $\phi \in [0, 2\pi]$ represent latitude and longitude, respectively. The associated Legendre polynomial P_ℓ^m in (1) is defined as:

$$P_\ell^m(x) = \frac{(-1)^m}{2^\ell \ell!} (1 - x^2)^{m/2} \frac{d^{\ell+m}}{dx^{\ell+m}} (x^2 - 1)^\ell \quad (2)$$

The SH functions Y_ℓ^m constitute a complete orthonormal system on a sphere. Any function $f(\theta, \phi)$ on a sphere can be expanded by a linear combination of these basis functions:

$$f(\theta, \phi) = \sum_{l=0}^{\infty} \sum_{m=-l}^l a_l^m Y_l^m(\theta, \phi), \quad (3)$$

where a_l^m is the coefficient of the basic function Y_l^m . Given a maximum degree ℓ_{\max} , an orthonormal system expanded by SHs involves $(\ell_{\max} + 1)^2$ coefficients. For a function with n spherical samples (θ_i, ϕ_i) and function values $f_i = f(\theta_i, \phi_i)$, $1 \leq i \leq n$ (i.e., f_i is the position of the sampled points of the point cloud or building model), coefficients a_l^m can be obtained by solving a least-square fitting [24]:

$$\begin{bmatrix} y_{1,1} & y_{1,2} & y_{1,3} & \cdots & y_{1,k} \\ y_{2,1} & y_{2,2} & y_{2,3} & \cdots & y_{2,k} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ y_{n,1} & y_{n,2} & y_{n,3} & \cdots & y_{n,k} \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_k \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_n \end{bmatrix}, \quad (4)$$

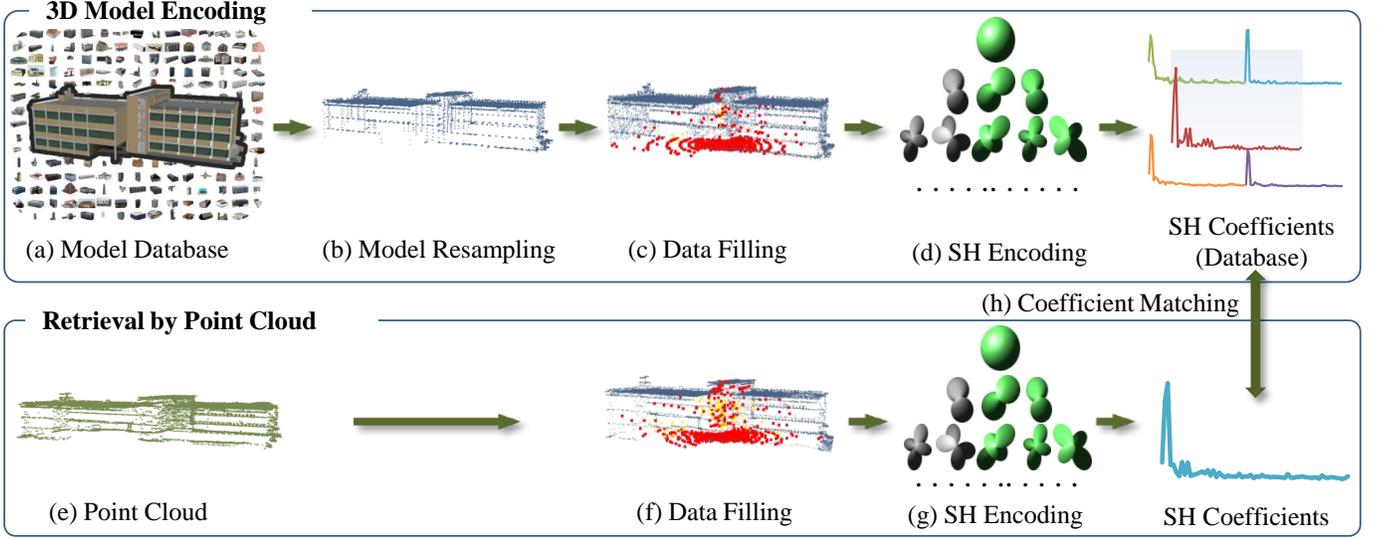


Fig. 1: System workflow. (a) 3D building model database; (b) surface resampling and (c) filling; (d) model encoding by SHs; (e) query input; (f) surface filling for point cloud; (g) point cloud encoding by SHs; and (h) coefficient matching and ranking.

where $y_{i,j} = Y_{\ell}^m(\theta_i, \phi_i)$, $b_j = a_{\ell}^m$, $j = \ell^2 + \ell + m + 1$ and $k = (\ell_{\max} + 1)^2$. To represent point clouds and building models by SHs, 3D geometric shapes are represented as spherical functions $f(\theta, \phi)$ in a polar coordinate system. The spherical function can be represented by three explicit functions $f(\theta, \phi) = (f_x(\theta, \phi), f_y(\theta, \phi), f_z(\theta, \phi))$, and the coefficients calculated by (4) are three-tuple vectors $a_{\ell}^m = (a_{\ell x}^m, a_{\ell y}^m, a_{\ell z}^m)$. By utilizing the fact that the L2-norm of SH coefficients are rotation invariant [25], the 3D geometric shape is encoded as follows:

$$SH(f(\theta, \phi)) = (\{\|a_{\ell}^m\|\}_{m=0}^0, \{\|a_{\ell}^m\|\}_{m=-1}^1, \dots, \{\|a_{\ell_{\max}}^m\|\}_{m=-\ell_{\max}}^{\ell_{\max}}) \quad (5)$$

This encoding is compact and has several useful properties for retrieval, which will be discussed in Section 4.1. Moreover, the 3D shape can be reconstructed using these coefficients with SHs:

$$f'(\theta, \varphi) = \sum_{\ell=0}^{\ell_{\max}} \sum_{m=-\ell}^{\ell} a_{\ell}^m Y_{\ell}^m(\theta, \varphi) \approx f(\theta, \varphi) \quad (6)$$

The 3D shapes approximated by SHs with different maximum degrees ℓ_{\max} are shown in Fig. 2. From this figure, we can see that the use of additional coefficients facilitates more accurate reconstruction. On the other hand, the use of fewer coefficients yields an effect similar to shape smoothing, meaning that the encoding is potentially insensitive to noise. In the experiments, the maximum degree ℓ_{\max} is set to 10 to account for noise insensitivity and efficient retrieval. Furthermore, since the coefficients a_{ℓ}^m are complex conjugates of the coefficients a_{ℓ}^{-m} , i.e., $\|a_{\ell}^m\| = \|-a_{\ell}^{-m}\|$, only coefficients a_{ℓ}^m with $m \geq 0$ (i.e., 66 coefficients) are used to encode the data for efficient retrieval and storage.

3.3 Consistent Data Encoding and Retrieval

As a preprocessing, the origins of the building models and the input point cloud are consistently set to the center of 3D object's bounding box. To consider the encoding efficiency, the simple axis-aligned bounding box is used rather than the minimum bounding box. The axis-aligned bounding box of an object represented by a set of points can be efficiently obtained by searching for the maximum point $(x_{\max}, y_{\max}, z_{\max})$ and minimum point $(x_{\min}, y_{\min}, z_{\min})$ in the Cartesian coordinate system. The center of bounding box, that is, $((x_{\max} - x_{\min})/2, (y_{\max} - y_{\min})/2, (z_{\max} - z_{\min})/2)$, is defined as the origin of model and point cloud. This process can reduce sensitivity to the 3D object origins in SH encoding. Besides, a model resampling process is performed by utilizing an airborne LiDAR simulator. The goal of this process is to make the sampling of building models similar to that of the point cloud acquired by airborne LiDAR, which can facilitate consistent encoding and accurate retrieval.

The proposed LiDAR simulator comprises two main components: *sensor* and *platform*. Generally, LiDAR sensors are mounted on positioning platforms such to determine the absolute positions and orientations of the sensors. Such platforms include a position and navigation system, which contains a GPS receiver and an INS. To simplify the simulator, the position and navigation system is excluded. Thus, a relative position is obtained rather than an absolute position. In the simulator, the sensor component is designed based on the fact that LiDAR measures the distance of a desired target by illuminating the target with light. Parameters for the sensor include laser pulse rate, i.e., scanning frequency, and scanning field of view (FOV), as illustrated in Fig. 3. The platform component is used to simulate the general

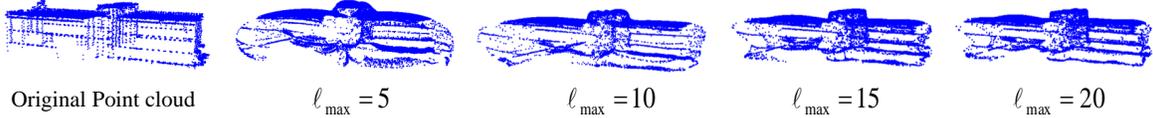


Fig. 2: 3D shape reconstruction. From left to right: original point cloud and SH reconstruction results using $l_{\max} = 5, 10, 15,$ and, 20 .

aircraft which includes the flight parameters: trajectory, height, and velocity. In the experiments, the laser pulse rate is set to 80 kHz (the range of laser pulse rate is generally 25 kHz to 100 kHz); the FOV is set to 40 degree; the flying height is set to 1000 m (the range of flying height is generally 500 m to 2000 m); the flying speed is set to 450 km/hr; and four strips surrounding the building model are set as the flight trajectory. In the simulation, the intersections of the building model and the rays emitted by the LiDAR simulator are calculated under the parameter settings of the sensor and platform, and a Gaussian noise is added to the sampled points to simulate the overall noises in the LiDAR system. Fig. 4 shows a model resampling generated by the proposed LiDAR simulator. The samplings of the building models are similar to those of the LiDAR point clouds, thus facilitating consistent encoding and successful model retrieval.

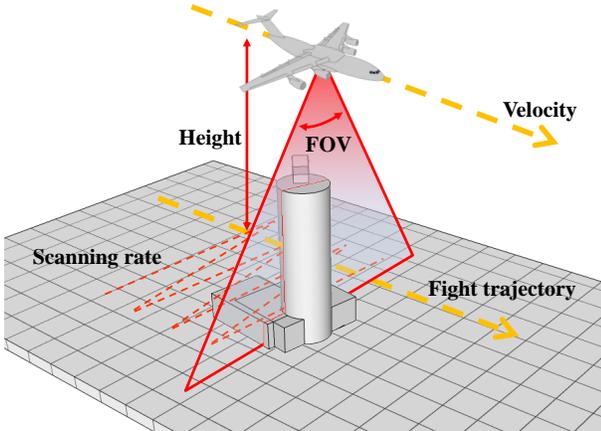


Fig. 3: Illustration of airborne LiDAR scanning. The yellow dotted line represents the flight trajectory, and the red dotted line represents the scanning path as well as the intersections between the 3D objects and the rays emitted by the LiDAR simulator.

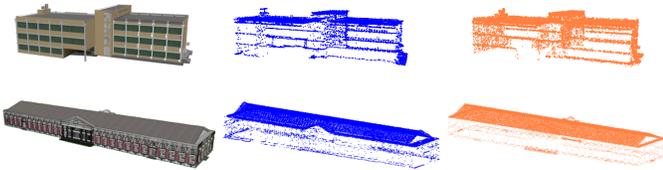


Fig. 4: Results of model resampling. From left to right: 3D building models, point clouds acquired by airborne LiDAR, and simulated point cloud generated by the LiDAR simulator.

Sparse and incomplete sampling is common and even inevitable in airborne LiDAR point clouds because of

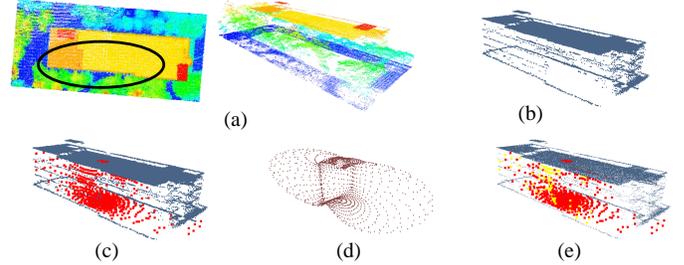


Fig. 5: Filling process. (a) Raw LiDAR point cloud in top and perspective views. Some parts of the building are occluded by neighboring trees (marked by black ellipse). The point cloud is visualized by point height colored ranging from green (lowest height) to red (highest height). (b) Building point cloud extracted from raw LiDAR data (a). (c) Filling the building ground and filling small holes by local interpolation. The sampled points are displayed by red. (d) SH reconstructed surface with $l_{\max} = 5$. (e) Filling with the aid of SH reconstructed surface. The sampled points are displayed by yellow.

obstacles (see the top figure in Fig. 5). Therefore, significant aliasing errors generally occur in encoded SH coefficients, which may result in retrieval failure. To solve this problem, a data filling process is required. Many approaches have been proposed for the surface interpolation and reconstruction of point clouds [26, 27]. These approaches mainly rely on the accurate point normal, which is a significant geometric feature of 3D shape. However, point normal estimation is nontrivial and difficult for unorganized, noisy, and incomplete point clouds. Therefore, the existing reconstruction approaches are infeasible for airborne LiDAR data. In this study, a simple and efficient data filling approach is proposed to insert several new point samples in order to meet the aliasing-free sampling constraint, that is, the sampling resolution must be larger than $180^\circ/l_{\max}$ [28]. The basic idea behind our approach is to insert samples within under-sampling regions by utilizing the existing points. This filling process consists of three steps: filling of building ground, filling of small and under-sampling regions by local interpolation, and filling of remaining holes, i.e., big holes, by utilizing an SH-reconstructed surface, as shown in Fig. 5. In the first step, several samples in the building ground are inserted because that the LiDAR rays cannot reach the building ground and the building ground information is missing. In the second step, an $n \times n$ grid mesh, where $n = 360/l_{\max}$, is created to cover the polar space (θ, ϕ) . The under-sampling grids are then efficiently determined in this space by verifying whether the grid is empty. To avoid extrapolation and

for reliable reconstruction, the empty and isolated grids are defined as small holes and linearly interpolated by the samples in their neighboring grids. In the third step, the interpolation of remaining empty grids, i.e., large empty regions, is guided by an approximated surface which is constructed by using the existing samples. The approximated surface is constructed using (6) with the encoded SH coefficients of the existing samples (using (4)). To generate a rough and approximated surface, ℓ_{\max} is set to 5. Several new point samples within the remaining under-sampling grids are then constructed using this approximated surface to meet the sampling criterion. Note that the values of maximum degree in gap filling ($\ell_{\max} = 5$) and data encoding ($\ell_{\max} = 10$) are different. Since that the gap filling fills the gaps based on the original point cloud only and the data encoding encodes data by using both the original point cloud and the new sampled points obtained from the gap filling, we assign a smaller value to ℓ_{\max} in the gap filling compared with that in the data encoding.

The experiment of point cloud encoding with and without the data filling process is shown in Fig. 6. The encoding results are compared with that of a simulated point cloud with dense sampling, which is obtained by uniformly sampling the corresponding building model. Obviously, without the data filling, the point clouds have significant errors on the encoded SH coefficients. By contrast, encoding with the process of data filling has SH coefficients similar to that of simulated point cloud, indicating an accurate encoding is obtained.

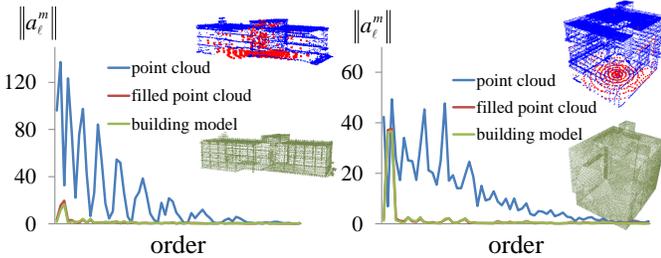


Fig. 6: Results of data filling and SH encoding. The blue, red, and green curves represent the encoded SH coefficients of the original point cloud, the filled point cloud, and the simulated point cloud (sampled from the corresponding building model), respectively.

3.4 Indexing and Ranking

SH encoding is inherently multi-resolutional. This property enables an efficient coarse-to-fine indexing and ranking. In this study, a three-stage ranking is used. The first stage uses a few low-frequency SH coefficients to define an initial search result. In the experiment, 21 SH coefficients (i.e., degree ≤ 5) are used, that is, $\{(\|a_0^m\|)_{m=0}^0, (\|a_1^m\|)_{m=0}^1, (\|a_2^m\|)_{m=0}^2, (\|a_3^m\|)_{m=0}^3, (\|a_4^m\|)_{m=0}^4, (\|a_5^m\|)_{m=0}^5\}$. The second stage uses the remaining coefficients to rank the resulting models within the returned model population. In other words, low-frequency coefficients define, in essence, the resulting

model population, and relatively high-frequency coefficients are used to rank the results. The overall shape similarity measurement is formulated as the distance between the encoded SH coefficients of the point cloud \mathbf{P} and the building model \mathbf{M} :

$$\text{dist}(\mathbf{P}, \mathbf{M}) = \|\text{SH}_i(\mathbf{P}) - \text{SH}_i(\mathbf{M})\|^2 \quad (7)$$

In the case of extracting the most identical building model for cyber city construction, the third stage is performed. In this stage, the standard registration algorithm called iterative closest point [29] is adopted to align the retrieved models, and then the building model that is the most identical to the query is extracted using the root mean square error (RMSE). In this manner, the building model that is the most close to the input point cloud is efficiently extracted from a database, and a city model containing the extracted building models can be efficiently constructed.

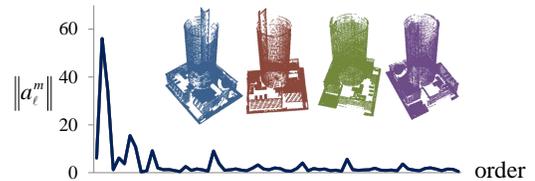


Fig. 7: Demonstration of the rotation invariance. Coefficients remain unchanged when rotations are applied to a point cloud.

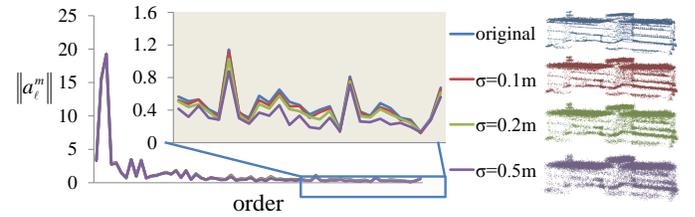


Fig. 8: Demonstration of the noise insensitivity. A point cloud with different Gaussian noise magnitudes is tested (the standard deviation σ is set to 0.1, 0.2, and 0.5 m). SH coefficients are slightly altered when the encoded data has noises.

4 EXPERIMENTAL RESULTS

4.1 Properties of the Proposed Encoding Approach

Shape retrieval based on the proposed encoding approach introduces several properties that demonstrate the potential for building model retrieval by point clouds. First, the proposed approach provides a metric in which similar shapes have small distances, whereas dissimilar ones have larger distances. Second, the proposed approach is capable of consistently encoding point clouds and polygon models with the aid of data resampling and filling. To demonstrate this property, building models and their corresponding point clouds are tested. The encoding results in Fig. 6 show that the building models and point clouds have similar coefficients, which indicates that they are consistently encoded. Third, the

coefficients are inherently rotation invariant. As shown in Fig. 7, the coefficients remain unchanged when rotations are applied to a building model. Fourth, the proposed encoding scheme is potentially insensitive to noise. It is because that only certain low-frequency SHs are used to encode the data, i.e., $l_{\max} = 10$. For example, in Fig. 8, a point cloud with different Gaussian noise magnitudes (the standard deviation is set to 0.1, 0.2, and 0.5 m) is tested. Results show that the encoded coefficients exhibit only slight differences when noise is present in the data. Fifth, SH encoding facilitates the reduction of dimensionality in shape description since a small set of SHs is used. With the aforementioned properties, the proposed retrieval method can efficiently and accurately retrieve polygon models by point clouds.

4.2 Parameter Setting

The maximal degree of SHs, i.e., l_{\max} , is the main parameter in the proposed encoding approach. To test the sensitivity and efficiency of model encoding to such parameter and to determine a suitable value, our approach was tested using various parameter values on simulated point clouds acquired from 3D polygon models. The experiment results are shown in Fig. 9. The blue curve denotes the encoding accuracy, which is calculated by measuring the differences between the 3D polygon models (the ground truths) and the SH reconstructed models. The red curve denotes the average computation time of the encoding. As expected, a tradeoff exists between the encoding accuracy and efficiency. A larger maximal degree corresponds to higher encoding accuracy but lower efficiency. Considering accuracy and efficiency, as well as noise insensitivity, the maximal degree of SHs is set to 10, that is, $l_{\max} = 10$. Note that it is difficult or even impossible to search for the optimal value of l_{\max} for all cases. Therefore, an empirical value is used in the implementation.

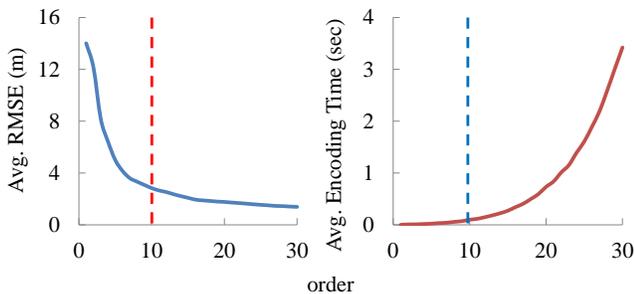


Fig. 9: Experiment of parameter setting. The blue curve denotes the encoding accuracy, and the red curve denotes the computation time of encoding. The simulated point clouds acquired by resampling polygon models are encoded using various values of l_{\max} .

4.3 Retrieval Evaluation and Application

Database and time complexity. Our approach was tested on a database containing approximately 15000 building

models mostly obtained from the Google 3D Warehouse. All experiments were evaluated on a PC with a 3.0 GHz GPU and 2 GB memory. Our system takes averagely 3.1 s to encode a point cloud with 10000 points and 0.85 s to search the database. Shape encoding is achieved by solving a least-squares system with an $n \times k$ design matrix, where n is the number of sampled points, and $k = (l_{\max} + 1)^2$. Thus, the computational complexity of shape encoding is $O(kn^2)$. If the retrieval is performed using sequential scanning without any index structure (the approach mentioned in Section 3.4), the computational complexity of retrieval is $O(N)$, where N is the number of building models in the database. This means that the time required to process a query is linearly dependent on the size of the model database.

Evaluation of the encoding scheme. The key to accurate retrieval is the consistency of building models and the corresponding LiDAR point cloud encoding. The existing model retrieval methods cannot well handle the point clouds because of the noises and incomplete shapes of point clouds. For instance, in the projection-based methods [5, 7, 10–12], a 3D model is represented as a set of 2D projections, and each projection is then represented by image descriptors. However, the 2D projections contain objects with sparse, noisy, and incomplete sampling, as shown in Fig. 10. The contour-based or region-based image descriptors may fail to encode the content. For the methods based on shape distribution [3, 6, 18], the histogram of 3D shape is used as the signature of 3D model. However, for a point cloud acquired by airborne LiDAR, the point sampling in the building roof is much denser than that in the side parts, as shown in Fig. 10. Besides, the point cloud is generally incomplete in shape. These problems make the methods difficult to generate a correct shape distribution.

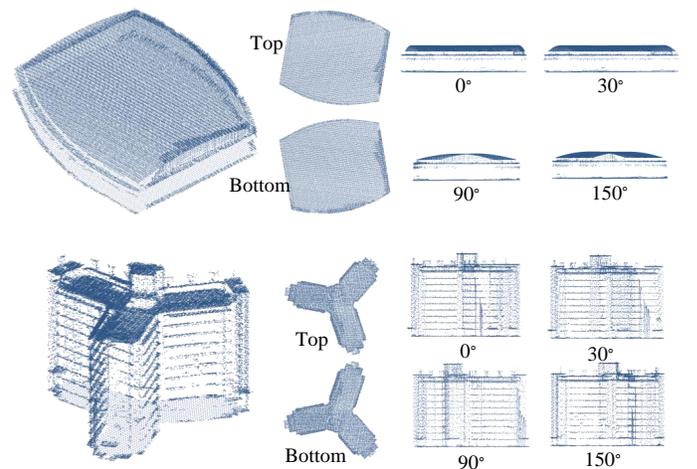


Fig. 10: Projection of point cloud. Left: point clouds. Right: various views of point clouds including top and bottom views, 0°, 30°, 90°, and 150° side views.

We propose the use of a LiDAR simulator in building model resampling to obtain a sampling that is similar to the LiDAR point cloud. To demonstrate the usability

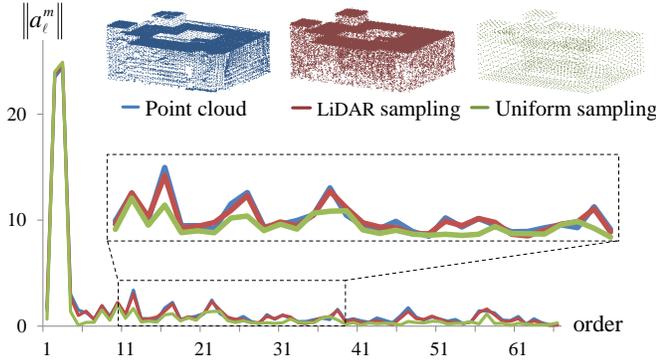


Fig. 11: Comparison between SH encodings of LiDAR point cloud, simulated point cloud from LiDAR simulator, and uniformly sampled point cloud.

of the LiDAR simulator, an encoding experiment using uniform sampling and the proposed sampling was conducted. The results are shown in Fig. 11. The proposed sampling is better, in terms of encoding accuracy, than uniform sampling for the building models that contains internal surfaces. The internal surfaces differentiate the uniform sampling from its corresponding LiDAR point cloud, resulting in inconsistent encoding. With the aid of LiDAR simulator, consistent sampling and encoding can be obtained. However, ambiguous encoding may occur because of the limitation of airborne LiDAR scanning. Parts of a building will at times be undetected by airborne LiDAR because the rays emitted by LiDAR are obstructed. This implies that the buildings with and without the undetected parts have similar encoded coefficients. For instance, the SH coefficients of a quadrangle building and that of a square building are similar, indicating the occurrence of ambiguous encoding. This is a limitation of airborne LiDAR point cloud encoding, and it is independent of the shape descriptor. The possible solution to this problem is to perform the third stage of ranking mentioned in Section 3.4. In this stage, the RMSE is used to refine the rankings further.

Retrieval evaluation. We compared our method with the related methods which are based on SH descriptor [2, 8] and shape distribution [3, 6, 18]. A dataset containing seven groups of building models shown in Fig. 12 were tested, and the commonly used measurements precision η_s and recall η_n were adopted to evaluate retrieval accuracy. 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. From the rankings and the precision-and-recall curves shown in Figs. 13 and 14, we conclude that the proposed method, which is capable of consistent encoding, is superior to the related methods. It should be note that the related studies mainly focused on the retrieval of polygon models by using a polygon model as input query. Therefore, these methods are unsuitable for the sparse, incomplete, and noisy LiDAR point clouds. As for the methods based on other descriptors such as shape topology [14], shape spectral [19], Zernike moment



Fig. 12: Model dataset. The dataset consists of seven groups (from top to bottom): Taipei 101 building, Shanghai world financial center building, apartment, square building, office building, rectangular store building, and other buildings.

[20], angular radial transform [21], and image signature [12], similar results will be obtained since the methods are designed for polygon models.

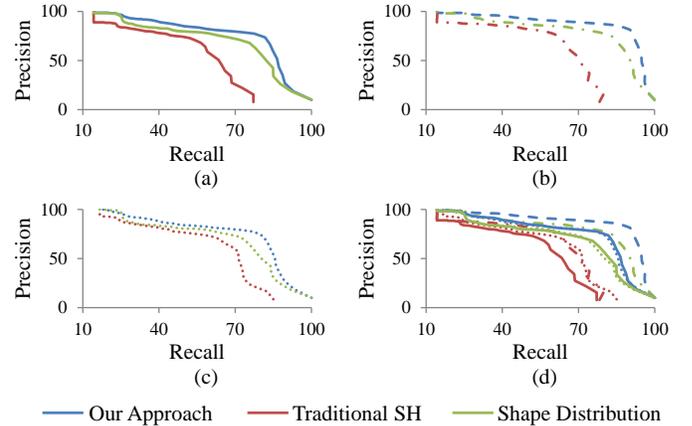


Fig. 14: Precision-and-recall curves of our method (blue) and the methods based on SH descriptor (red) and shape distribution (green). The dataset shown in Fig. 12 is tested. The curves are the average precision and recall for (a) all groups, (b) without the group of Shanghai world financial center building, and (c) without the group of other buildings. (d) Combining the average precision and recall curves shown in (a)-(c).

To demonstrate the feasibility of the proposed method, a huge database containing approximately 15000 building models was tested. The retrieval and ranking results shown in Fig. 15 indicate that similar models are retrieved and most of the retrieved models have correct rankings. However, incorrect rankings still occur. For this problem, similarly, the third stage of ranking can be performed, and the rankings can be further refined



Fig. 13: Retrieval results. Retrieval and ranking results generated by the traditional SH (top), shape distribution (middle), and our method (bottom). The leftmost model is the query, and the model dataset shown in Fig. 12 is tested.

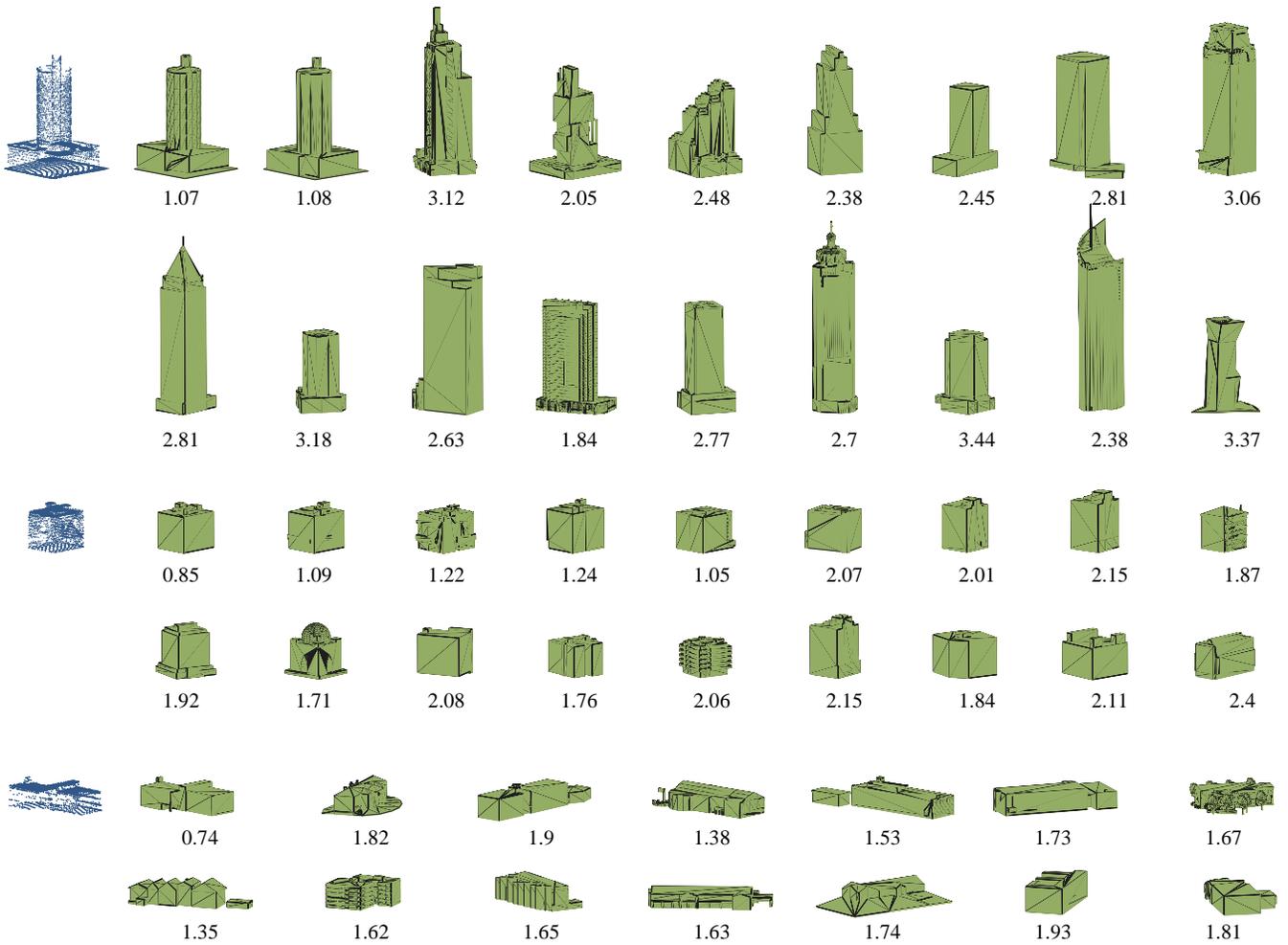


Fig. 15: Retrieval from a huge model database. Left: query input (point clouds acquired by LiDAR). Right: ranking results with RMSE.

by RMSE.

Application. With the aid of our system, users can efficiently construct a cyber campus/city, as shown in Fig. 16. The ground objects are scanned by airborne LiDAR, and the point clouds of the buildings are extracted using the classification technique [30]. With our system, a building model in the database that is most close to the query is retrieved, and a cyber campus containing the retrieved models can be efficiently constructed.

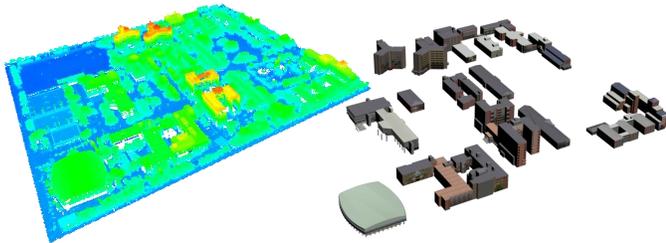


Fig. 16: Cyber campus construction. Left: LiDAR point cloud. The height is visualized by color ranging from blue to red. Right: retrieved building models.

5 CONCLUSIONS AND FUTURE WORK

A building model retrieval method using a point cloud as query input was presented. With the proposed encoding approach, the building models in the database and the input point clouds can be consistently encoded. The proposed encoding approach based on SHs introduces the properties of rotation invariance and noise insensitivity. Moreover, such an approach provides a compact and hierarchical encoding mechanism, which reduces search time and repository space requirements. Besides, the data resampling and filling have been proven to mitigate the encoding difficulties that arise from the sparse and incomplete sampling of point clouds. The experimental results of airborne LiDAR data demonstrate the efficiency and accuracy of the proposed approach, and the qualitative and quantitative analyses show the clear superiority of the proposed method over the related methods. In addition, a cyber campus generated by retrieving 3D building models with airborne LiDAR point clouds demonstrates the feasibility of our approach. In the near future, we plan to develop an approach that retrieves tree models with point clouds, thus enabling the construction of a more complete cyber city model.

REFERENCES

- [1] J. P. Wilson, "Digital terrain modeling," *Geomorphology*, vol. 137, issue 1, pp. 107-121, 2012.
- [2] T. Funkhouser, P. Min, M. Kazhdan, J. Chen, A. Halderman, D. Dobkin, and D. Jacobs, "A search engine for models," *ACM Trans. on Graphics*, vol. 22, no. 1, pp. 83-105, 2003.
- [3] Jürgen Assfalg, Marco Bertini, Alberto Del Bimbo, and Pietro Pala, "Content-based retrieval of 3-D objects using spin image signatures," *IEEE Trans. on Multimedia*, vol. 9, no. 3, pp. 589-599, 2007.
- [4] A. Mademlis, P. Daras, D. Tzovaras, and M. G. Strintzis, "Ellipsoidal harmonics for 3-D shape description and retrieval," *IEEE Trans. on Multimedia*, vol. 11, issue 8, pp. 1422-1433, 2009.
- [5] Y. Gao, M. Wang, Z.-J. Zha, Q. Tian, Q.-H. Dai, and N.-Y. Zhang, "Less is more: efficient 3-D object retrieval with query view selection," *IEEE Trans. on Multimedia*, vol. 13, issue 5, pp. 1007-1018, 2011.
- [6] C. B. Akgul, B. Sankur, Y. Yemez, and F. Schmitt, "3D model retrieval using probability density-based shape descriptors," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 31, no. 6, pp. 1117-1133, 2009.
- [7] Y. Gao, J. Tang, R. Hong, S. Yan, Q. Dai, N. Zhang, and T.-S. Chua, "Camera constraint-free view-based 3D object retrieval," *IEEE Trans. on Image Processing*, vol. 21, no. 4, pp. 2269-2281, 2012.
- [8] M. Kazhdan, T. Funkhouser, and S. Rusinkiewicz, "Rotation invariant spherical harmonic representation of shape descriptors," *Proc. Eurographics/ACM SIGGRAPH Symp. Geometry Processing*, pp. 156-164, 2003.
- [9] R. C. Veltkamp, T. Haar, and B. Frank, "Shape retrieval contest (SHREC)," *IEEE International Conference on Shape Modeling and Applications*, pp. 215-216, 2008.
- [10] D. Y. Chen, X. P. Tian, and Y. T. Shen, "On visual similarity based 3D model retrieval," *Computer Graphics Forum*, vol. 22, no. 3, pp. 223-232, 2003.
- [11] P. Papadakisa, I. Pratikakisa, S. Perantonisa, and T. Theoharis, "Efficient 3D shape matching and retrieval using a concrete radialized spherical projection representation," *Pattern Recognition*, vol. 40, no. 9, pp. 2437-2452, 2007.
- [12] G. Stavropoulos, P. Moschonas, K. Moustakas, D. Tzovaras, and M. G. Strintzis, "3-D model search and retrieval from range images using salient features," *IEEE Trans. on Multimedia*, vol. 12, no. 7, pp. 692-704, 2010.
- [13] M. Hilaga, Y. Shinagawa, and T. Kohmura, "Topology matching for fully automatic similarity estimation of 3D shapes," *ACM Trans. on Graphics*, pp. 203-212, 2001.
- [14] K. L. Tam and W. H. Lau, "Deformable model retrieval based on topological and geometric signatures," *IEEE Trans. on Visualization and Computer Graphics*, vol. 13, no. 3, pp. 470-482, 2007.
- [15] S. Biasotti, D. Giorgi, M. Spagnuolo, and B. Falcidieno, "Size functions for comparing 3D models," *Pattern Recognition*, vol. 41, no. 9, pp. 2855V2873, 2008.
- [16] M.-W. Chao, C.-H. Lin, C.-C. Chang, T.-Y. Lee, "A graph-based shape matching scheme for 3D articulated objects," *Computer Animation and Virtual Worlds*, vol. 22, no. 2-3, pp. 295-305, 2011.
- [17] A. Mademlis, P. Daras, A. Axenopoulos, D. Tzovaras, and M. G. Strintzis, "Combining topological and geometrical features for global and partial 3D

- shape retrieval," *IEEE Trans. on Multimedia*, vol. 10, no. 5, pp. 819V831, 2008.
- [18] R. Osada, T. Funkhouser, B. Chazelle, and D. Dobkin, "Shape distributions," *ACM Trans. on Graphics*, vol. 21, no. 4, pp. 807-832, 2002.
- [19] V. Jain and H. Zhang, "A spectral approach to shape-based retrieval of articulated 3D models," *Computer-Aided Design*, vol. 39, no. 5, pp. 398V407, 2007.
- [20] M. Novotni and R. Klein, "Shape retrieval using 3D zernike descriptors," *Computer-Aided Design*, vol. 36, no. 11, pp. 1047V1062, 2004.
- [21] J. Ricard, D. Coeurjolly, and A. Baskurt, "Generalizations of angular radial transform for 2D and 3D shape retrieval," *Pattern Recognition Letters*, vol. 26, no. 14, pp. 2174-2186, 2005.
- [22] P. Daras, D. Zarpalas, D. Tzovaras, and M. G. Srintzsis, "Efficient 3-D model search and retrieval using generalized 3-D radon transforms," *IEEE Trans. Multimedia*, vol. 8, no. 1, pp. 101V114, Feb. 2006.
- [23] P. Shilane, K. Michael, M. Patrick, and T. Funkhouser, "The princeton shape benchmark," *International Conference on Shape Modeling*, pp. 167-178, 2004.
- [24] C. Brechbuhler, G. Gerig, and O. Kubler, "Parametrization of closed surfaces for 3-D shape description," *Computer Vision and Image Understanding*, vol. 61, no. 2, pp. 154-170, 1995.
- [25] L. Shen and M. K. Chung, "Large-scale modeling of parametric surfaces using spherical harmonics," *3DPVT 06: Proc. Third Intl Symp. 3D Data Processing, Visualization, and Transmission*, pp. 294-301, 2006.
- [26] M. Kazhdan, M. Bolitho, and H. Hoppe, "Poisson surface reconstruction," *Proceedings of the fourth Eurographics symposium on Geometry processing*, pp. 61-70, 2006.
- [27] I-C. Yeh, C.-H. Lin, O. Sorkine and T.-Y. Lee, "Template-based 3D model fitting using dual-domain relaxation," *IEEE Trans. on Visualization and Computer Graphics*, vol. 17, pp. 1178-1190, 2011.
- [28] T.-H. Li and G. North, "Aliasing effects and sampling theorems of spherical random fields when sampled on a finite grid," *Annals of the Institute of Statistical Mathematics*, vol. 49, no. 2, pp. 341-364, 1997.
- [29] P. J. Besl, and N. D. McKay, "A method for registration of 3-D shapes," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 14, no. 2, pp. 239-256, 1992.
- [30] A. P. Charaniya, R. Manduchi, and S. K. Lodha, "Supervised parametric classification of aerial LiDAR data," *Proceedings of the IEEE 2004 Conference on Computer Vision and Pattern Recognition Workshop*, pp. 1-8, 2004.