Received May 12, 2018, accepted June 6, 2018, date of publication June 13, 2018, date of current version June 29, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2846759

An Approach to 3D Building Model Retrieval Based on Topology Structure and View Feature

TAO LIU[1,2], JIN GAO[3], AND YUCHEN ZHAO[2]
[1]State Key Laboratory of CAD&CG, Zhejiang University, Hangzhou 310058, China
[2]College of Big Data, North University of China, Taiyuan 030051, China
[3]China Waterborne Transport Research Institute, Beijing 100088, China

Corresponding author: Tao Liu (dlmult@hotmail.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 61602426, Grant 61672473, and Grant 61702147, in part by the China Postdoctoral Science Foundation under Grant 2017M621932, and in part by the Science Foundation of North University of China under Grant XJJ2016031.

ABSTRACT This paper proposes an approach to 3-D building model retrieval based on a topology structure and view feature using two filter steps to finish the building model retrieval. First, the distance transformation method is used to extract the topological structure of 3-D building models. The obtained skeleton points of 3-D building models are classified by our concentric sphere method. The 3-D building models are then filtered for a first time based on the distribution features. Second, 3-D building models are projected onto 2-D images from the viewpoints of skeleton points. The scale-invariant feature transform algorithm is used to extract feature points from 2-D projection images, and the idea of the bag of feature method is used to compare the projection images. The 3-D building models are then filtered for a second time. Finally, we can get the retrieval result (the matched models). This method first carries out the matching of the overall structure and then carries out the matching of local details. This conforms to the cognitive style of human. In addition, the experiment results show the effectiveness of our method.

INDEX TERMS 3D building model retrieval, topology structure, view feature, distance transformation, SIFT algorithm.

I. INTRODUCTION

With the development of virtual reality, 3D GIS and other 3D technologies, there is an explosion in the number of 3D building models and there is an increasing demand for the research of 3D building model retrieval. There are trillions of 3D models on the Internet. How to effectively acquire the required 3D models has become an urgent problem. It will take a large time cost to build a high precision 3D model. Therefore, in order to reuse existing models, the research of analyzing, matching and retrieving 3D models becomes very important. 3D model retrieval provides a method for us to acquire models conveniently and fast. In this way, we can reuse the 3D model resources. Most of 3D building models have features of obvious edge, simple topological structure and distinct style. This can obviously improve the efficiency of 3D building model retrieval.

Many algorithms have already been proposed to extract the feature of 3D models. These algorithms are generally classified into five categories: view-based method, graph-based method, geometry-based method, surface attribute-based method, and deep learning method. All these methods have their own advantages and limitations. The view-based method has a high discriminative power and a high robustness. It does not require pose normalization. But these methods could not support partial matching. Atmosukarto and Shapiro [1] present a 3D object retrieval method using salient views. Daras and Axenopoulos [2] provide a method to extract descriptors from 2D projection images of 3D models to finish 3D model retrieval. Papadakis *et al.* [3] obtain a panoramic view of a 3D object by projecting it to the lateral surface of a cylinder parallel to one of its three principal axes and centered at the centroid of the object. Then the corresponding 2D Discrete Fourier Transform and 2D Discrete Wavelet Transform are performed to retrieve 3D models. Shih *et al.* [4] construct a novel feature, called elevation descriptor, for 3D model retrieval. The descriptor is invariant to scaling and translation of models and it is robust for rotation. Mahmoudi and Daoudi [5] design a 3D search engine based on characteristic views of 3D model and a probabilistic Bayesian voting method.

Curvature Scale Space (CSS) descriptor is used to get the characteristic views. Gao *et al.* [6] introduce a view-based 3D model retrieval method using probabilistic graph model. Each captured view set is modeled as a first order Markov Chain. Yasseen *et al.* [7] present a new approach for sketch-based 3D object retrieval that describes a 2D shape by the visual protruding parts of its silhouette. They also propose criteria for locating side, off axis, or asymmetric views. The graph-based method also has a good discriminative power. It does not require pose normalization and has the ability to perform partial matching. Robustness of these methods is weaker compared with that of the view-based method. Amenta *et al.* [8], [9] first calculate the voronoi diagram of 3D models, and then they calculate the skeleton model based on the voronoi diagram to describe the global feature of 3D model. Inspired by the existing works, retrieval methods based on the topological structure are proposed. Ma and Wan [10] present an algorithm based on the multi-resolution reeb graph. Cornea *et al.* [11] compute hierarchical curve-skeletons of 3D objects. Hilaga *et al.* [12] propose a topological matching method which uses Reeb graphs based on a quotient function defined by an integral geodesic distance. Zuckerberger *et al.* [13] apply an approach similar to model graphs to content based retrieval. They decompose the model surface into patches and identify adjacent patches to build a graph representation of the model. Sundar *et al.* [14] use as a shape descriptor a skeletal graph that encodes geometric and topological information. Iyer *et al.* [15] use global features and skeletal graphs to describe volume models, obtained by voxelizing solid models. Skeleton method does not need to normalize the 3D model. And applicable for partial matching. The geometry-based method is a large category of model retrieval methods, which mainly has four sub-categories: global feature, global feature distribution, spatial map, local feature. Discriminative power of these methods is not good. But these methods have a high robustness. For most methods, pose normalization is required. And except for local feature methods, a drawback of geometry-based methods is that partial matching is not supported. Osada *et al.* [16] present a distribution feature according to the relationship between the vertexes of different geometric surfaces. Zaharia and Preteux [17] propose the method of 3D Shape Spectrum Descriptor (3DSSD) to describe 3D models. Mahmoudi and Daoudi [18] present a method to obtain the curvature distributional feature of 3D models extracted from the curvature of surface vertex. This algorithm calculates the space distribution of vertex curvatures of seven 2D projection feature views obtained from the 3D model. Gal and Cohen-Or [19] construct a method for partial matching of surfaces represented by triangular meshes. This method matches surface regions that are numerically and topologically dissimilar, but approximately similar regions. Kuo and Cheng [20] propose a 3D shape representation scheme based on a combination of principal plane analysis and dynamic programming. Lü *et al.* [21] use a method combining a distance histogram and moment invariants to improve the retrieval performance and propose a mutual information distance measurement to perform the similarity comparison of 3D objects. Gao *et al.* [22] propose a 3D model comparison algorithm based on a 3D model descriptor: spatial structure circular descriptor (SSCD). SSCD can preserve the global spatial structure of 3D models, and is invariant to rotation and scaling. Zou *et al.* [23] construct a novel combined shape distribution (CSD) descriptor for 3D model retrieval based on principal plane analysis and group integration. The surface attribute-based method uses surface attributes of 3D model, such as color, reflection coefficient and texture, to perform the content based retrieval. Paquet and Rioux [24] separately consider the color and photometric characteristics of surface material. They use the color histogram of each component of the RGB color space to describe the color properties of the surface material and use seven different histograms to describe the texture features. Suzuki [25] proposes an interactive 3D model color retrieval method. The material color is expressed as the integrated value of several illumination parameters in the illumination model, and a multiple regression analysis method is used to predict the illumination model. There is not much research in this area. Most studies are biased towards the extraction of geometry, topology, and view features. In recent years, there are some researchers use deep learning methods to do the model retrieval [26], [27]. Zhu *et al.* [28] propose to learn a robust domain-invariant representation between 3D shape and depth image domains by constructing a pair of discriminative neural networks, one for each domain. Yang *et al.* [29] design and implement a constructive-learning for cross-model correlation algorithm for 3D model retrieval. Both the vertex correlation and the edge correlation are simultaneously learned in the constructive-learning process. However, these researches need a large amount of training computation. Combining different approaches may produce more powerful shape matching methods in the future.

During the construction of 3D GIS systems, building design systems and virtual city systems, the scene construction is the most time consuming stage. It usually takes up 80% to 90% of the overall construction cost. At present, people have accumulated a lot of building model resources through various model construction methods. How to reuse the existing assets for new 3D systems becomes an important issue. And 3D building model retrieval becomes a very valuable research topic. At present, there are few targeted researches on the 3D building model retrieval and there is no open 3D building model data set. In addition, few retrieval methods focus on the study of integrated structure and view feature retrieval of 3D models. Single feature retrieval model can hardly provide high efficiency for 3D building model retrieval.

3D building models have the feature of simple topology structure and they have the topology invariance with respect to translation, rotation, and scaling. And 3D building models have significant texture feature. This is especially true for city scene applications. Texture feature can quickly distinguish between different models, such as office buildings and castles. And texture feature can distinguish between similarly structured but completely different models, such as cubes and office buildings. We propose a 3D building model retrieval method by combining topology structure and texture feature. We first compare the skeleton structures of different 3D building models, and then their 2D projection images obtained from the viewpoints of skeleton points are compared. The goal of the second operation is to distinguish models with similar geometry structure, for example a box and a skyscraper. This is critical for the 3D building model retrieval. Through these two similarity measurements, we can get the retrieval result. And the experiment shows the effectiveness of our approach.

II. BASIC METHODS AND CONCEPTS

The skeleton can retain the topological features of an object to the maximum extent. Skeleton extraction is a process of simplifying an object. Curve skeleton extraction algorithms are mainly divided into two categories. One is based on the thinning method [30], [31]. Calculation steps obey the condition of topological invariance and the criterion of neighborhood information of voxels. The iterative method is adopted to eliminate the ordinary points step by step until only skeleton points are left. However, the skeleton produced by thinning algorithm can hardly be smooth and accurate. The other is based on the distance transform algorithm [32]. This kind of algorithms performs the distance transformation to find the skeleton points of an object. It is difficult to guarantee the connectivity of the skeleton when branches are very thin and small. However, 3D building models have very few important thin and small branches and their shapes are relatively regular. So, we use the distance transform algorithm to extract the curve skeleton of 3D models in our algorithm.

We use three steps to extract the skeleton points of 3D building models:

(1). Voxelize the 3D building model.

(2). Perform the distance transformation.

(3). Extract the skeleton points.

A. 3D MODEL VOXELIZATION

Before extracting the skeleton of 3D models, we should first obtain the voxel representation of models. Voxelization transforms the surface geometry representation of a 3D model into the voxel representation and produces volume datasets which include not only surface information of the model but also the internal attribute information. This process has two main steps: boundary voxelization and interior voxelization. We use octree to speed up the calculation in this paper. The detailed steps of this process are described as follows:

(1). Calculate minimum bounding box for the 3D building model and create the coordinate system. Firstly, we should find the minimum vertex $(x_{\min}, y_{\min}, z_{\min})$ and the maximum vertex $(x_{\max}, y_{\max}, z_{\max})$ in the vertex list of the mesh model and construct a bounding box with the two points as the diagonal vertex. Then, we create a voxel coordinate system. The coordinate system is: the coordinate origin is the minimum vertex $(x_{\min}, y_{\min}, z_{\min})$, the coordinates are integer coordinates, the X-axis, Y-axis and Z-axis are parallel to X-axis, Y-axis and Z-axis of the Euclidean space, respectively. The voxel space is a 3D discrete space and a subspace of 3D Euclidean space. Each integer sequence (i, j, k) uniquely determines a voxel. At last, we define a variable called the voxel zone bit. The voxel zone bit $f(i, j, k)$ satisfies the following functional relationships: If the voxel is outside the boundary of the object, then $f(i, j, k) = -1$; If the voxel is on the boundary of the object, then $f(i, j, k) = 0$; If the voxel is inside the boundary of the object, then $f(i, j, k) = 1$.

(2). Build octree to organize the data. Octree structure has been widely used in computer graphics, computer vision, image processing, and so on. It is an important way to describe the 3D model. The principle is discussed in many literatures [33], [34]. Usually, octree just decomposes the "gray node". In this paper, we decompose all nodes and evenly divide the bounding box of 3D building model along the axis direction in the voxel space.

Meagher proposes the linear octree encoding scheme. The octree node is encoded as $q_1 q_2 \cdots q_n$. And the coding bit q_l should be equal to $w_l v_l u_{l(2)}(4{*}w_l+2{*}v_l+u_{l(10)})$, $w_l, v_l, u_l \in (0, 1)$. Voxel coordinate (i, j, k) in the voxel space is calculated according to (1).

$$\begin{cases} i = u_1 \times 2^{n-1} + u_2 \times 2^{n-2} + \cdots + u_l \times 2^{n-l}; \\ j = v_1 \times 2^{n-1} + v_2 \times 2^{n-2} + \cdots + v_l \times 2^{n-l}; \\ k = w_1 \times 2^{n-1} + w_2 \times 2^{n-2} + \cdots + w_l \times 2^{n-l}. \end{cases} \tag{1}$$

Where n is the depth of the octree and l is the current level.

Therefore, we can calculate the linear node coding based on the voxel coordinate (i, j, k) of each octree node. The binary component of each coding bit is calculated according to (1). Each coding bit is calculated according to (2). At last, we will get the node coding.

$$q_l = w_l v_l u_{l(2)}(4w_l + 2v_l + u_{l(10)}). \tag{2}$$

(3). Voxelize the boundary. We first voxelize the surface of 3D building model. The model surface is voxelized to obtain the voxel contour of 3D model. 3D model surface is represented by polygon meshes (usually triangle meshes). This process is divided into three calculations: point voxelization, edge voxelization and triangle face voxelization. During the calculation, octree structure is used to perform the voxel neighborhood search.

For the point $P(x, y, z)$, we first calculate the local grid coordinates of the voxel to which the point P belongs according to (3).

$$\begin{cases} Locijk[0] = floor(x/cellx); \\ Locijk[1] = floor(y/celly); \\ Locijk[2] = floor(z/cellz). \end{cases} \tag{3}$$

Where *cellx*, *celly*, *cellz* are the geometric scales of each voxel unit along the X, Y, and Z axes.

Based on the local grid coordinates of the voxel, we can calculate the binary component of the voxel to which the point P belongs according to (1). And we can calculate the voxel coding according to (2). At last, we will get the place of the voxel to which the point P belongs through the coding, and set the value of the corresponding voxel zone bit to 0.

For the edge E, we should first get the relationship between E and the coordinate planes. If the edge E is perpendicular to one coordinate plane, it indicates that this edge is parallel to one coordinate axis. So, one coordinate component of voxels of the edge E increases or decreases in the direction of the axis and the other two coordinate components are equal to the corresponding coordinate components of the voxel of the start point. If the edge E is parallel to one coordinate plane $P\pi$, it indicates that one coordinate component of voxels of the edge E is constant. The constant value is the coordinate component of the start point along the axis that is perpendicular to the coordinate plane $P\pi$. Planar scan-conversion method is used to calculate the other two coordinate components by projecting the edge E onto the coordinate plane $P\pi$. If the edge E is neither perpendicular nor parallel to one coordinate plane, ray tracing method is used to calculate the voxels that the edge passes through [33].

For the triangle face, the calculation can be divided into two steps. Firstly, project the triangle face onto the coordinate plane which has the maximum projected area and use planar scan method to calculate the two projected coordinate components of interior voxels based on the voxel coordinates of vertices and edges. Then, calculate the voxel coordinates of intersection points between the triangle face and rays constructed by the two projected coordinate components along the third axis.

(4). Voxelize the interior. After the voxelization of the boundary of 3D model has been finished, a method similar to the seed filling algorithm, called flooding operation, is used to voxelize the interior of model. Firstly, initialize the voxel zone bit of all voxels to 1, and then set the voxel zone bit of boundary voxels to 0 based on the boundary voxelization and set the voxel zone bit of the minimum voxel $V(x_{\min}, y_{\min}, z_{\min})$ and the maximum voxel $V(x_{\max}, y_{\max}, z_{\max})$ to -1. Secondly, set the voxel zone bit of a voxel to -1 if its adjacent voxel's voxel zone bit is -1 during the forward and backward scan. At last, the voxel zone bits of all surface voxels are set to 0, the voxel zone bits of all interior voxels are set to 1 and the voxel zone bits of all exterior voxels are set to -1. Execution speed of this approach is very fast. It is only related to the size of the volume data and independent of the complexity of the model.

B. 3D DISTANCE TRANSFORMATION

Since the concept of Digital Distance Transformation (DDT) was first proposed by Rosenfield and Pfalt in 1966, DDT has been widely used to compute distances from points to feature points in image processing, computer vision, and so on. DDT offers the distance field calculation [35].

In the 2D space, an $n \times n$ binary image can be considered to contain only two kinds of pixels: feature pixel and background pixel. We call the feature pixel ''black spot'' and call the background pixel ''white spot''. In the distance transformation image, the value of each pixel represents the distance from the pixel to the nearest black spot. In the actual calculation, there are two kinds of distance measures: non-Euclidean distance and Euclidean distance. Among them, the non-Euclidean distance transformation is an approximation of the Euclidean distance transformation and cannot obtain the accurate position of the skeleton point of the object. Therefore, we adopt the Euclidean distance transformation in this paper. 3D Euclidean distance transformation algorithm can be regarded as the 3D extension of 2D Euclidean distance transformation. In the 3D space, 3D Euclidean distance transformation is performed on the models represented by voxels. 3D model represented by polygon meshes should be voxelized. The transformation can be calculated as follows:

Firstly, decompose the $n \times n \times n$ 3D binary image into n $n \times n$ 2D binary images, and then perform the 2D Euclidean distance transformation on each $n \times n$ 2D binary image to calculate the nearest black spot for each pixel in the 2D binary image.

Secondly, calculate and compare the distance between pixels in each 2D binary image and black spots in the other 2D binary images based on the results of 2D Euclidean distance transformation.

Finally, use an optimization method to reduce the number of 2D binary images and black spots involved in the distance calculation and comparison [36].

C. EXTRACTION OF SKELETON POINTS

When we get the distance field $DT(x, y, z)$ of a 3D model based on the 3D Euclidean distance transformation, then we can extract the skeleton points of the model. In general, the position of skeleton points obtained by the distance transformation-based skeleton extraction algorithm is relatively accurate. Distance field obtained by the distance transformation can be seen as a scalar field, and the associated vector field is the gradient of the distance transformation. In this paper, we use distance field and its gradient to calculate the skeleton points of a model. There is a relationship between the distance field and the model skeleton: the distance transformation value of skeleton point is larger than that of most of its adjacent points. Suppose $DT(x_0, y_0, z_0)$ represents the distance transformation value of voxel point (x_0, y_0, z_0). The gradient of distance field can be expressed by (4).

$$\nabla DT = (\frac{\partial DT}{\partial x}, \frac{\partial DT}{\partial y}, \frac{\partial DT}{\partial z}). \tag{4}$$

The norm of gradient vector is calculated by (5).

$$|\nabla DT| = \sqrt{(\frac{\partial DT}{\partial x})^2 + (\frac{\partial DT}{\partial y})^2 + (\frac{\partial DT}{\partial z})^2}. \tag{5}$$

In order to calculate skeleton points of the model, we first need to define the local maximum point of distance transformation. Suppose p is a voxel of the voxel model, $DT(p)$ is the distance transformation of p, Q is the set of adjacent points of p. If $\forall q \in Q$ such that $DT(p) \geq DT(q)$, then the point p is called the local maximum point of distance transformation. As mentioned above, the distance transformation value of skeleton point should be larger than that of most of its adjacent points. It can be concluded that the local maximum point of distance transformation is a potential skeleton point. Here we give the definition of the key point. Key point is a local maximum point which has the minimum $|\nabla DT|$ value among adjacent local maximum points, and key points are connected. A group of connected local maximum points corresponds to a convex part of an object and the smaller $|\nabla DT|$ value indicates the existence of skeleton point. So, we choose the key point as the skeleton point.

The application effect of this method is shown in Fig. 1. Fig. 1(a) and Fig. 1(c) show the original 3D building models. Fig. 1(b) displays the skeleton points of the 3D building model in Fig. 1(a) and Fig. 1(d) displays the skeleton points of the 3D building model in Fig. 1(c).

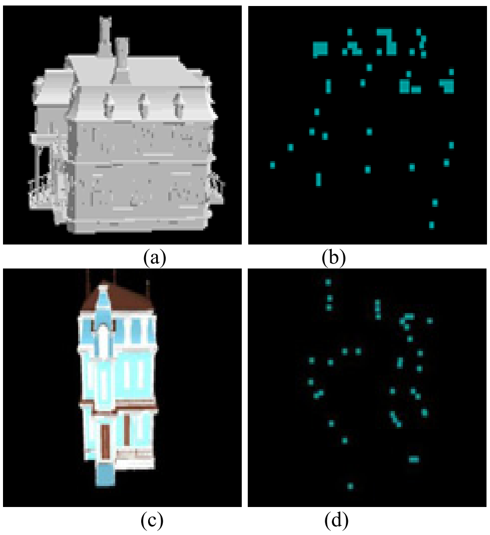

FIGURE 1. The schematic diagram for skeleton points of the 3D building model. (a) Original 3D building model 1. (b) Skeleton points of the 3D building model in (a). (c) Original 3D building model 2. (d) Skeleton points of the 3D building model in (c).

III. 3D BUILDING MODEL RETRIEVAL METHOD

In this paper, we propose a 3D building model retrieval algorithm based on topological structure and view feature. This algorithm mainly includes two matching processes: topological structure matching and view feature matching. Topological structure matching is used to quickly identify building models with different styles or structures and view feature matching is used to identify building models in the local detail. This process conforms to the cognitive style of human. Fig. 2 shows the flow chart of our algorithm.

A. MODEL MATCHING BASED ON TOPOLOGICAL STRUCTURE

In order to distinguish different 3D building models, we first classify skeleton points of the model to extract the distribution feature of skeleton points and perform the first matching according to the distribution feature. The comparison result can be sorted according to the similarity. We use topological structure to identify building models with different styles or structures, such as villa and office building. We design a concentric sphere method to extract the distribution feature of skeleton points of the 3D building model. The detailed steps are described as follows:

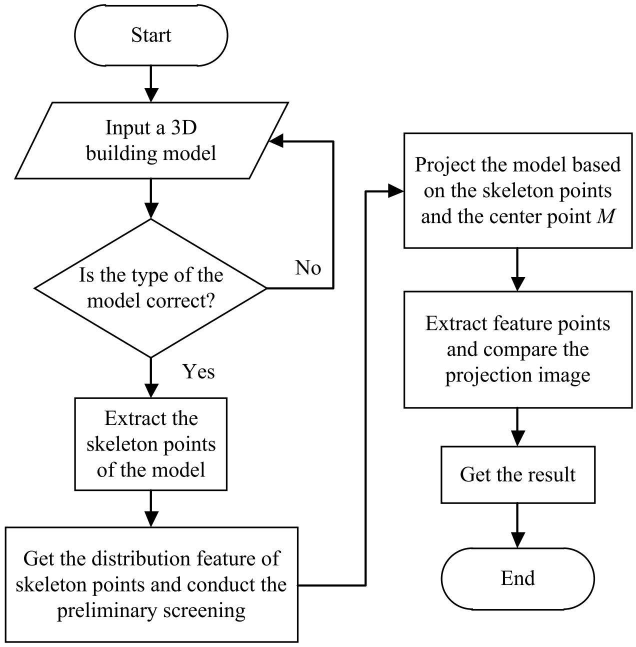

FIGURE 2. The flow chart of 3D building model retrieval method.

Firstly, use the method described in Section 2.3 to extract the skeleton points of 3D building models.

Secondly, calculate the geometric center of the 3D building model. Suppose $D = \{(x_1, y_1, z_1), (x_2, y_2, z_2), \cdots, (x_n, y_n, z_n)\}$ is the coordinate point array of the 3D building model, then the geometric center point M of the 3D building model can be calculated by (6). M is the center of gravity of the 3D building model. That is, we use the regional volume distribution as an attribute to measure the irregularity of the model.

$$M = (x_0 = \frac{\sum_{k=1}^{n} x_n}{n}, y_0 = \frac{\sum_{k=1}^{n} y_n}{n}, z_0 = \frac{\sum_{k=1}^{n} z_n}{n}). \tag{6}$$

Thirdly, construct the distance classification set. Suppose $S = \{(i_1, j_1, k_1), (i_2, j_2, k_2), \cdots, (i_n, j_n, k_n)\}$ is the skeleton point set of the 3D building model. We can calculate a set of distances between skeleton points and the geometric center point M according to (7).

$$E = \left\{ \begin{array}{l} p_1 = \sqrt{(x_0 - i_1)^2 + (y_0 - j_1)^2 + (z_0 - k_1)^2}, \\ p_2 = \sqrt{(x_0 - i_2)^2 + (y_0 - j_2)^2 + (z_0 - k_2)^2}, \cdots, \\ p_n = \sqrt{(x_0 - i_n)^2 + (y_0 - j_n)^2 + (z_0 - k_n)^2} \end{array} \right\}. \tag{7}$$

We calculate the maximum value p_{m} of the set E by traversing all elements in the set, and then construct the distance classification set $O = \left\{ \frac{p_1}{p_m}, \frac{p_2}{p_m}, \cdots, \frac{p_n}{p_m} \right\}$ base on the set E.

Finally, classify the distance classification set to get the distribution feature of skeleton points. We first establish a series of concentric spheres to classify the elements in the set O. These spheres are centered at point M. The radius of the outmost sphere is 1. Then, we count the numbers of points falling into different spheres to get the distribution feature of skeleton points. The distribution feature has a central symmetry property, and is independent of geometrical model transformations such as rotation, translation, and scaling. This method is sensitive to the number of concentric spheres. We experimentally test the effect of this parameter on the result many times and the experiment results show that the result is relatively better when the radius of the outmost sphere is divided into five equal parts. So, five concentric spheres will be better. Fig. 3 shows the distribution histogram of skeleton points of the 3D building model in Fig. 1(c).

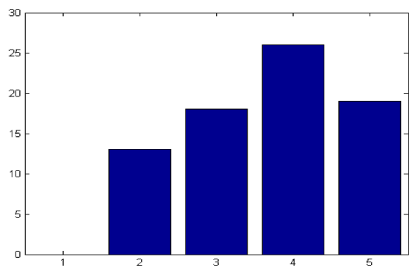

FIGURE 3. The distribution histogram of skeleton points of the 3D building model in Fig. 1(c).

Through the comparison of the distribution feature of skeleton points of the target 3D building model and distribution features of skeleton points of the 3D building model in the database, we can conduct a preliminary screening to filter out most part of 3D building models in the database. The detailed comparison process and parameter settings are as follows. The comparison threshold of two corresponding bars is set to 0.1. Here the comparison threshold is the ratio of error to the original value. Its value can be adjusted according to varying application case. If the calculated value is no bigger than the threshold, then these two bars are considered similar. If the number of similar bars is no less than 3, then the two histograms are considered similar, that is, the two corresponding 3D building models are similar. And the matching result can be sorted according to the number of similar bars.

The application effect of this matching method is shown in Fig. 4. Fig. 4(a) shows the calculated histograms. Histograms 1, 2, 3, 4, 5, 6, 7 correspond to Fig. 4(b), Fig. 4(c), Fig. 4(d), Fig. 4(e), Fig. 4(f), Fig. 4(g) and Fig. 4(h) respectively. Fig. 4(b) displays the input target 3D building model. Fig. 4(c), Fig. 4(d), Fig. 4(e), Fig. 4(f), Fig. 4(g) and Fig. 4(h) display the retrieval results.

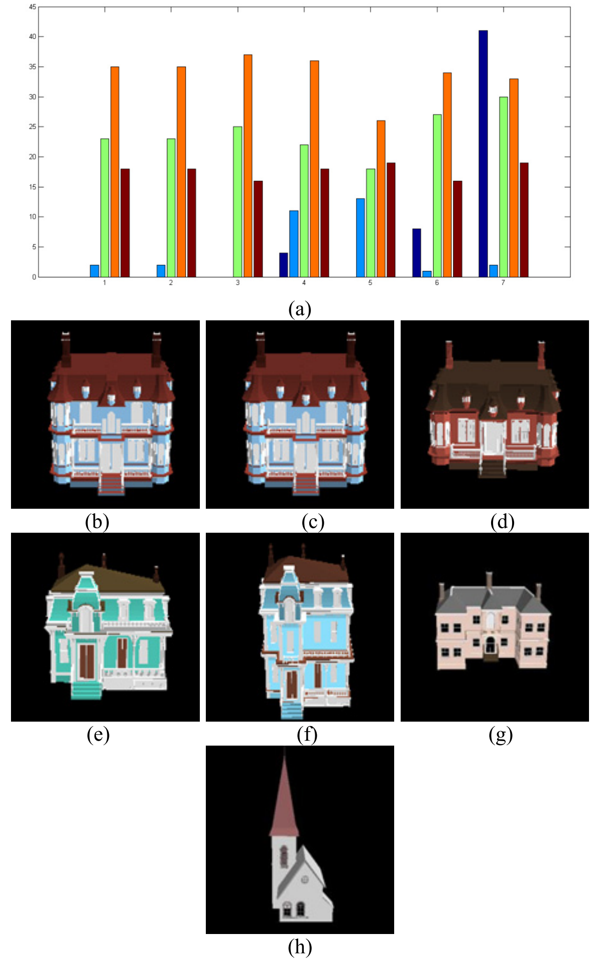

FIGURE 4. The schematic diagram for model matching based on topological structure. (a) Distribution histogram. (b) Input target 3D building model. (c) Retrieval result 1. (d) Retrieval result 2. (e) Retrieval result 3. (f) Retrieval result 4. (g) Retrieval result 5. (h) Retrieval result 6.

B. MODEL MATCHING BASED ON VIEW FEATURE

After the first matching, we continue to match the remaining 3D building models for the second time. The second model matching based on view feature is a filter-refinement process based on the first model matching. This process is mainly divided into three steps: model projection, feature point extraction and projection image comparison.

1) MODEL PROJECTION

We have got the skeleton points of the 3D building model during the first matching. Then, we project 3D building models onto 2D images from the viewpoints of skeleton points to obtain the 2D projection images. The projection plane is perpendicular to the line passing through the skeleton point and the geometric center point M. Fig.5 shows the projection images of the 3D building model. The first row denotes some projection images of the 3D building model in Fig. 4(c), and the second row denotes some projection images of the 3D building model in Fig. 4(d).

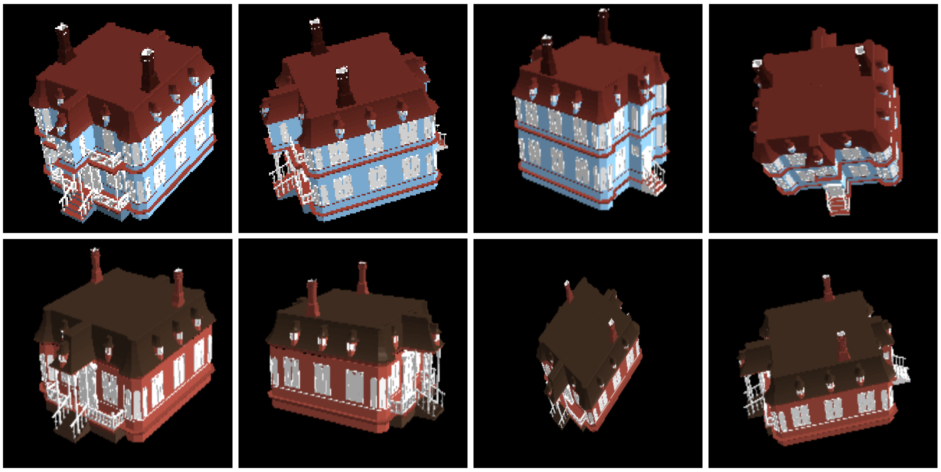

FIGURE 5. The schematic diagram for projection images of the 3D building model.

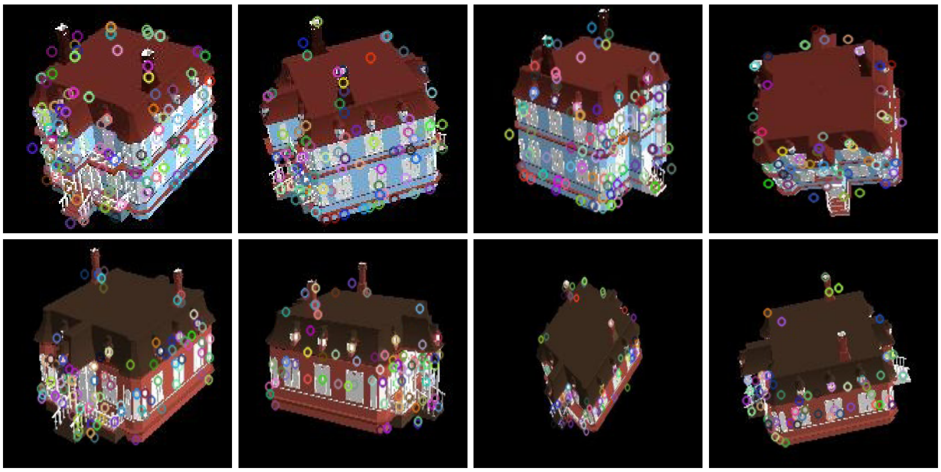

FIGURE 6. SIFT feature points of projection images.

2) FEATURE POINT EXTRACTION

We use Scale Invariant Feature Transform (SIFT) algorithm [37] to extract the feature points of projection images. The SIFT algorithm extracts local features of the image. It is invariant to image translation, rotation, scaling and noise. The SIFT descriptor represents the gradient statistical result of neighborhood Gaussian image of the feature point. SIFT has unparalleled advantages in the invariant feature extraction of images and is suitable for fast and accurate matching in massive feature databases. Firstly, gradient calculation is performed for the feature point. Then, histogram is used to count the gradient and direction of each pixel in the feature point's neighborhood. We divide the direction range of 0 to 360 degrees into 18 directions, and the peak point of direction histogram is used as the direction of gradients in the feature point's neighborhood. So, feature points of the projection image can be divided into 18 categories according to the amplitude and direction of the gradient. Based on our experiments, we recommend using a classification number of 18. In our experiment, the classification number of 18 is sufficient. The retrieval result increases only slightly when the classification number is above 18. In our calculation framework, the classification number is an experimental parameter. Certainly, in different applications, it can be set as other values.

Fig. 6 shows the feature points of projection images extracted by the SIFT algorithm. The first row denotes the projection images of the 3D building model in Fig. 4(c), and the second row denotes the projection images of the 3D building model in Fig. 4(d). Circles denote the feature points extracted by the SIFT algorithm.

3) PROJECTION IMAGE COMPARISON

Bag of Feature (BOF) method uses the view feature word to describe an image [38]. View feature word is the cluster center of local features. The local feature can be calculated by LBP descriptor, SIFT descriptor or SURF descriptor. The set of view feature words of an image is vividly called the bag of feature.

FIGURE 7. The schematic diagram for the application effect of our algorithm.

The main idea of BOF method is to represent images by a set of local features rather than by a global image description. We use feature points extracted by the SIFT algorithm to represent the 2D projection images of 3D building models. Based on the feature point extraction operation described above, feature points of the projection image are divided into 18 categories. Each category is used as a view feature word in this paper. And we use view feature word histogram to compare the similarity between two projection images.

The detailed comparison process and parameter settings are as follows. We firstly divide each projection image into 8 patches. Based on our experiments, we recommend using a patch number of 8. Then, the view feature word histogram is constructed for each patch by counting the number of feature points in each category. The comparison threshold of two corresponding bars is set to 0.8. Its value can be adjusted according to different application cases. If the calculated value is no bigger than the threshold, then these two bars are considered similar. If the number of similar bars is no less than 9 (half the number of bars), then the two histograms are considered similar, that is, the two corresponding patches are similar. If the number of similar patches is no less than 4 (half the number of patches), then the two projection images are considered similar.

Finally, we use this image comparison method to retrieve remaining 3D building models for the second time. For a 3D building model, if the number of similar projection images between this model and the target model is no less than half the number of projection images of the target model, then this model is similar to the target model. And the matching result can be sorted according to the number of similar projection images.

IV. EXPERIMENT AND RESULTS

Because Princeton ModelNet and ShapeNet provide few 3D building models [39], [40], we use 225 3D building models which are created by ourselves or obtained from the Internet and 100 similar 3D models, such as boxes, building block models, and so on, to build an experimental dataset. These 3D models are represented by the triangle mesh. We use this dataset to test our algorithm. Firstly, we input a target 3D building model. Then, topological structure-based model matching method is used to perform the first filter and view feature-based model matching method is used to match the remaining 3D building models for the second time. We use Precision-Recall curve to describe the retrieval performance of different retrieval methods.

Fig. 7 shows the application effect of our algorithm. The first column denotes the input target models, and the other columns denote the retrieved models. Fig. 8 displays the Precision-Recall curves for 3D building model retrieval methods based on different features. It can be clearly concluded that our method is better than single feature retrieval methods.

We compare our method with the multi-view CNN retrieval algorithm [26] and the method based on component space distribution [41], as shown in Fig. 9. It can be seen from the Recall-Precision curves that our algorithm shows better performance than the other two algorithms in the building model database. The experiment results show the effectiveness and efficiency of our method.

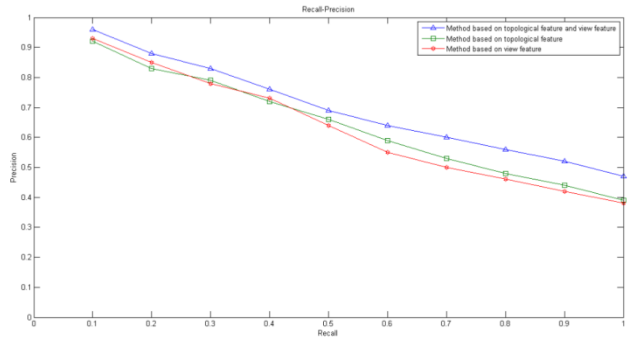

FIGURE 8. **Precision-Recall curves for 3D building model retrieval methods based on different features.**

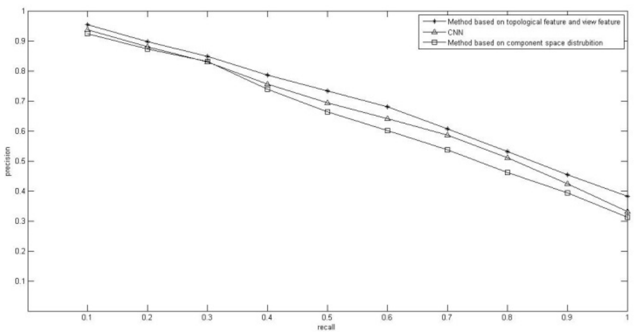

FIGURE 9. **Precision-Recall curves for different 3D building model retrieval methods.**

The most time consuming step of our method is the skeleton point extraction of the 3D building model. Time complexity of this step is $O(n^3 \log n)$, where n is the voxelization resolution of the model. Time complexity of the feature point extraction of the projection image is $O(n^2)$, where n is the resolution of the projection image. In our experiment, the skeleton point extraction calculation takes on average 30ms and the average time of the feature extraction calculation for 3D building model is 56ms. The time complexity of the forward feeding portion of CNN network is $O(n^3)$, where n is the number of inputs. The time complexity grows linearly with the number of the layer, which is $O(p*n)$, where n is the number of layers and p is the average number of perceptrons in each layer. And the time complexity of the back propagation portion is $O(p*n)$. In our experiment, the training of the network is the most time consuming stage. This takes 4.685 minutes. The network takes on average 865ms for one model retrieval. The time complexity of the core calculation of component space distribution method is $O(n\log n)$, where n is the number of the model vertex. In our experiment, this method takes on average 10ms for the feature calculation of a 3D building model. In summary, component space distribution method has the highest computational efficiency, followed by our method. The multi-view CNN method has the lowest computational efficiency.

V. CONCLUSION

This paper proposes a hybrid 3D building model retrieval method based on topology structure and view feature. Unlike CAD models, building models have significant texture features. This method combines the advantages of topology feature method and view feature method. It firstly compares the topology feature of 3D building model based on the calculation of the distribution feature of skeleton points, and then compares the view feature of 3D building model based on the calculation of the distribution feature of SIFT feature points. These view feature points are extracted from 2D projection images, and the extraction calculation is based on the topology feature of 3D building model. It can be seen from the experiment result that our method is better than single feature retrieval methods and some other methods.

Topology structure and view feature of the model are self-related attributes. So, the extraction calculation of topology structure and view feature can be precomputed. And these features can be stored in the database as model attributes. In the real-time application, we can directly use these attributes to do the comparison calculation. In our calculation framework, specific parameter values need to be adjusted according to specific application requirements. And, we need to compromise between computational efficiency and accuracy.

However, the calculation of the topological structure of input 3D building model is relatively complex. This will take a lot of time during the real-time retrieval. Further work has to be done in terms of computation optimization. We will combine the GPU rasterization, the LOD method and an early rejection strategy to construct a more generic method and test models with small branches.

REFERENCES

[1] I. Atmosukarto and L. G. Shapiro, "3D object retrieval using salient views," *Int. J. Multimedia Inf. Retrieval*, vol. 2, no. 2, pp. 103–115, Jun. 2013.

[2] P. Daras and A. Axenopoulos, "A 3D shape retrieval framework supporting multimodal queries," *Int. J. Comput. Vis.*, vol. 89, nos. 2–3, pp. 229–247, 2010.

[3] P. Papadakis, I. Pratikakis, T. Theoharis, and S. Perantonis, "PANORAMA: A 3D shape descriptor based on panoramic views for unsupervised 3D object retrieval," *Int. J. Comput. Vis.*, vol. 89, nos. 2–3, pp. 177–192, Sep. 2010.

[4] J.-L. Shih, C.-H. Lee, and J. T. Wang, "A new 3D model retrieval approach based on the elevation descriptor," *Pattern Recognit.*, vol. 40, no. 1, pp. 283–295, 2007.

[5] S. Mahmoudi and M. Daoudi, "A probabilistic approach for 3D shape retrieval by characteristic views," *Pattern Recognit. Lett.*, vol. 28, no. 13, pp. 1705–1718, Oct. 2007.

[6] Y. Gao, J. Tang, H. Li, Q. Dai, and N. Zhang, "View-based 3D model retrieval with probabilistic graph model," *Neurocomputing*, vol. 73, nos. 10–12, pp. 1900–1905, 2010.

[7] Z. Yasseen, A. Verroust-Blondet, and A. Nasri, "View selection for sketch-based 3D model retrieval using visual part shape description," *Vis. Comput.*, vol. 33, no. 5, pp. 565–583, May 2017.

[8] N. Amenta, S. Choi, and R. K. Kolluri, "The power crust, unions of balls, and the medial axis transform," *Comput. Geometry*, vol. 19, nos. 2–3, pp. 127–153, Jul. 2001.

[9] N. Amenta, S. Choi, and R. K. Kolluri, "The power crust," in *Proc. 6th ACM Symp. Solid Modeling Appl.*, Ann Arbor, MI, USA, 2001, pp. 249–266.

[10] C.-M. Ma and S.-Y. Wan, "Parallel thinning algorithms on 3D (18, 6) binary images," *Comput. Vis. Image Understanding*, vol. 80, no. 3, pp. 364–378, Dec. 2000.

[11] N. D. Cornea, D. Silver, X. Yuan, and R. Balasubramanian, "Computing hierarchical curve-skeletons of 3D objects," *Vis. Comput.*, vol. 21, no. 11, pp. 945–955, Oct. 2005.
[12] M. Hilaga, Y. Shinagawa, T. Kohmura, and T. L. Kunii, "Topology matching for fully automatic similarity estimation of 3D shapes," in *Proc. 28th ACM Annu. Conf. Comput. Graph. Interact. Techn.*, New York, NY, USA, 2001, pp. 203–212.
[13] E. Zuckerberger, A. Tal, and S. Shlafman, "Polyhedral surface decomposition with applications," *Comput. Graph.*, vol. 26, no. 5, pp. 733–743, Oct. 2002.
[14] H. Sundar, D. Silver, N. Gagvani, and S. Dickinson, "Skeleton based shape matching and retrieval," in *Proc. IEEE Shape Modeling Int.*, Seoul, South Korea, May 2003, pp. 130–139.
[15] N. Iyer, Y. Kalyanaraman, K. Lou, S. Jayanti, and K. Ramani, "A reconfigurable 3D engineering shape search system: Part I—Shape representation," in *Proc. ASME Int. Design Eng. Tech. Conf. Comput. Inf. Eng. Conf.*, Chicago, IL, USA, 2003, pp. 89–98.
[16] R. Osada, T. Funkhouser, B. Chazelle, and D. Dobkin, "Matching 3D models with shape distributions," in *Proc. Int. Conf. Shape Modeling Appl.*, Genova, Liguria, Italy, 2001, pp. 154–166.
[17] T. Zaharia and F. J. Preteux, "3D-shape-based retrieval within the MPEG-7 framework," *Proc. SPIE*, vol. 4304, pp. 133–145, May 2001, doi: 10.1117/12.424969.
[18] S. Mahmoudi and M. Daoudi, "3D models retrieval by using characteristic views," in *Proc. 16th IEEE Int. Conf. Pattern. Recognit.*, Quebec City, QC, Canada, Aug. 2002, pp. 457–460.
[19] R. Gal and D. Cohen-Or, "Salient geometric features for partial shape matching and similarity," *ACM Trans. Graph.*, vol. 25, no. 1, pp. 130–150, Jan. 2006.
[20] C. T. Kuo and S. C. Cheng, "3D model retrieval using principal plane analysis and dynamic programming," *Pattern Recognit.*, vol. 40, no. 2, pp. 742–755, Feb. 2007.
[21] K. Lü, N. He, and J. Xue, "Content-based similarity for 3D model retrieval and classification," *Prog. Natural Sci.*, vol. 19, no. 4, pp. 495–499, Apr. 2009.
[22] Y. Gao, Q. Dai, and N. Y. Zhang, "3D model comparison using spatial structure circular descriptor," *Pattern Recognit.*, vol. 43, no. 3, pp. 1142–1151, Mar. 2010.
[23] K.-S. Zou, W.-H. Ip, C.-H. Wu, Z.-Q. Chen, K.-L. Yung, and C.-Y. Chan, "A novel 3D model retrieval approach using combined shape distribution," *Multimed. Tools Appl.*, vol. 69, no. 3, pp. 799–818, Apr. 2014.
[24] E. Paquet and M. Rioux, "A content-based search engine for VRML databases," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Santa Barbara, CA, USA, Jun. 1998, pp. 541–546.
[25] M. T. Suzuki, "A web-based retrieval system for 3D polygonal models," in *Proc. 20th NAFIPS Int. Conf. IFSA World Congr.*, Vancouver, BC, Canada, 2001, pp. 2271–2276.
[26] C. R. Qi, H. Su, M. Nießner, A. Dai, M. Yan, and L. J. Guibas, "Volumetric and multi-view cnns for object classification on 3D data," in *Proc. 29th IEEE Conf. Comput. Vis. Pattern Recognit.*, Las Vegas, NV, USA, Jun. 2016, pp. 5648–5656.
[27] P. Wang, Y. Liu, Y. Guo, C. Sun, and X. Tong, "O-CNN: Octree-based convolutional neural networks for 3D shape analysis," *ACM Trans. Graph.*, vol. 36, no. 4, Jul. 2017, Art. no. 72.
[28] J. Zhu, J. R. Rizzo, and Y. Fang, "Learning domain-invariant feature for robust depth-image-based 3D shape retrieval," *Pattern Recognit. Lett.*, pp. 1–10, Oct. 2017, doi: 10.1016/j.patrec.2017.09.041.
[29] J. Yang, J. Zhao, and Q. Sun, "3D model retrieval using constructive-learning for cross-model correlation," *Neurocomputing*, vol. 275, no. 31, pp. 1–9, Jan. 2018.
[30] D. Y. Chen and M. Ouhyoung, "A 3D object retrieval system based on multi-resolution Reeb graph," in *Proc. Comput. Graphic. Workshop (Taiwan)*, Tainan, Taiwan, 2002, pp. 16–20.
[31] W. Xie, R. P. Thompson, and R. Perucchio, "A topology-preserving parallel 3D thinning algorithm for extracting the curve skeleton," *Pattern Recognit.*, vol. 36, no. 7, pp. 1529–1544, Jul. 2003.
[32] J. Liu, W. Liu, C. Wu, and L. Yuan, "A new method of extracting objects'curve-skeleton," (in Chinese), *Acta Automatica Sinica*, vol. 34, no. 6, pp. 617–622, Jun. 2008.
[33] X. Wu, W. Liu, and T. Wang, "Octree structure based voxelization of polygonal meshes," (in Chinese), *J. Eng. Graph.*, vol. 26, no. 4, pp. 1–7, Aug. 2005.
[34] F. Liu, Z. Li, C. Li, and Z. Cao, "An linear octree based model for visualization of mining processes," in *Proc. 8th IEEE Int. Conf. Fuzzy Syst. Knowl. Discovery*, Shanghai, China, Jul. 2011, pp. 2470–2474.
[35] X. Wu, W. Liu, and M. Y. Wang, "Feature based modeling of heterogeneous objects," *Multidiscipline Model. Mater. Struct.*, vol. 1, no. 4, pp. 341–366, Jul. 2005.
[36] Y. Zhuge, J. Tian, and W. Wang, "A new method of three-dimensional Euclidean distance transform," (in Chinese), *J. Softw.*, vol. 12, no. 3, pp. 383–389, Mar. 2001.
[37] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, 2004.
[38] Y. Liu, H. Zha, and H. Qin, "Shape topics: A compact representation and new algorithms for 3D partial shape retrieval," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, New York, NY, USA, Jun. 2006, pp. 2025–2032.
[39] K. V. Vishwanath, D. Gupta, A. Vahdat, and K. Yocum, "ModelNet: Towards a datacenter emulation environment," in *Proc. IEEE 9th Int. Conf. Peer-Peer Comput.*, Seattle, WA, USA, Sep. 2009, pp. 81–82.
[40] A. X. Chang *et al.* (Dec. 2015). "ShapeNet: An information-rich 3D model repository." [Online]. Available: https://arxiv.org/abs/1512.03012
[41] L. Wan, Q. Zhao, and A. Hao, "A method of 3D model retrieval by the spatial distributions of components," (in Chinese), *J. Softw.*, vol. 18, no. 11, pp. 2902–2913, Nov. 2007.

TAO LIU was born in Linyi, China in 1988. He received the M.S. and Ph.D. degrees in traffic information engineering and control from Dalian Maritime University, Dalian, China, in 2011 and 2015, respectively. He currently holds a post-doctoral position at the State Key Lab of CAD&CG, Zhejiang University, China. He is involved in the area of computer graphics. His current research interests include real-time rendering, model feature analysis, and 3-D display techniques.

JIN GAO was born in Dezhou, China in 1989. He received the B.S. degree in communication engineering from Hainan University, Haikou, China, in 2011, and the M.S. degree in traffic information engineering and control from Dalian Maritime University, Dalian, China, in 2013. He is currently an Engineer with the China Waterborne Transport Research Institute, China. His current research interests include transport simulation and 3-D visualization.

YUCHEN ZHAO was born in Yuncheng, China in 1993. He received the B.S. degree in computer science and technology from Hainan University, Haikou, China, in 2015. He is currently pursuing the M.S. degree in computer application technology with the North University of China, China. His research interests include visualization simulation.