Hindawi
Complexity
Volume 2020, Article ID 9050459, 14 pages
https://doi.org/10.1155/2020/9050459

Research Article

An Efficient 3D Model Retrieval Method Based on Convolutional Neural Network

Bo Ding, Lei Tang, and Yong-jun He

School of Computer Science and Technology, Harbin University of Science and Technology, Harbin 150080, China

Correspondence should be addressed to Yong-jun He; holywit@163.com

Received 3 February 2020; Accepted 19 May 2020; Published 11 June 2020

Academic Editor: Qingling Wang

Recently, 3D model retrieval based on views has become a research hotspot. In this method, 3D models are represented as a collection of 2D projective views, which allows deep learning techniques to be used for 3D model classification and retrieval. However, current methods need improvements in both accuracy and efficiency. To solve these problems, we propose a new 3D model retrieval method, which includes index building and model retrieval. In the index building stage, 3D models in library are projected to generate a large number of views, and then representative views are selected and input into a well-learned convolutional neural network (CNN) to extract features. Next, the features are organized according to their labels to build indexes. In this stage, the views used for representing 3D models are reduced substantially on the premise of keeping enough information of 3D models. This method reduces the number of similarity matching by 87.8%. In retrieval, the 2D views of the input model are classified into a category with the CNN and voting algorithm, and then only the features of one category rather than all categories are chosen to perform similarity matching. In this way, the searching space for retrieval is reduced. In addition, the number of used views for retrieval is gradually increased. Once there is enough evidence to determine a 3D model, the retrieval process will be terminated ahead of time. The variable view matching method further reduces the number of similarity matching by 21.4%. Experiments on the rigid 3D model datasets ModelNet10 and ModelNet40 and the nonrigid 3D model dataset McGill10 show that the proposed method has achieved retrieval accuracy rates of 94%, 92%, and 100%, respectively.

1. Introduction

Recently, three-dimensional (3D) models have been widely used in computer-aided design (CAD), virtual reality (VR), 3D animation and film, medical diagnosis, 3D online games, machinery manufacturing, and other fields. In particular, with the development of 3D printing, the application of 3D models has become an indispensable technical means in all fields. Since more and more 3D models and digitizing tools are being developed for an ever-increasing number of applications, a large number of 3D models have become available on the Web [1]. Through the Internet, users can download free 3D models according to their needs. Modification and incremental design on these models can not only reduce product cost and shorten design time, but also effectively improve product reliability and quality. However, it is very difficult to find the needed 3D model quickly and accurately from the massive number of available models. 3D model retrieval techniques can solve the above problems; therefore, this technique has become a research hotspot.

One important issue of the 3D model retrieval is to represent models into descriptors. The descriptors describe the 3D model accurately and efficiently to support model classification, index building, and similarity matching. 3D model descriptors can be mainly divided into four categories: geometry-based [2], statistical analysis-based [3], topology-based [4], and projective view-based descriptors [5]. For the geometry-based 3D model descriptors, the 3D model is divided into many grids, and then the features of the 3D model are extracted by different mathematical transformations of the grid model. The earliest work on the former approach is the 3D ShapeNets [6], which learns a convolutional deep belief network that outputs probability distributions of binary occupancy voxel values. After that,

Maturana and Scherer propose a similar approach, which builds the VoxNet for real-time object recognition [7]. Li et al. adopt field probing neural networks (FPNNs) to extract features of 3D models. In this method, the 3D models are first represented as volumetric fields, and then the field probing filters are employed to extract features from them [8]. Wu et al. propose a novel framework named the 3D Generative Adversarial Network (3D-GAN), which generates 3D objects from a probabilistic space by leveraging recent advances in volumetric convolutional networks and generative adversarial nets. This method achieves impressive performance on 3D object recognition [9].

Statistical analysis-based 3D model descriptors are a good choice for nonrigid 3D model retrieval. The earliest work is the 3D rotation invariant spherical harmonic representation of 3D shape descriptors (SHP), which reduces the dimensionality of the descriptor and provides a more compact representation [10]. Sun et al. propose the heat kernel signature (HKS) descriptor to describe the local characteristics of the nonrigid 3D models. It is based on diffusion scale-space analysis and characterized by the heat transfer process of the 3D surface [11]. The HKS descriptor is invariant under isometric deformations and stable under perturbations of the model. It has achieved good performance in nonrigid 3D model retrieval. However, it is sensitive to the scale changes of the 3D model. Aubry et al. propose the wave kernel signature (WKS) descriptor to describe the nonrigid 3D model, which describes the average probability of quantum mechanics at a position on a nonrigid 3D model surface. The WKS descriptor explains the relationship between the points on the different spatial scales and the rest of the model surface, and its discriminative ability is more than that of the HKS descriptor [12]. Zeng et al. use WKS and HKS to represent the 3D model, then construct two convolutional neural networks for the HKS distribution and the WKS distribution separately, and use the multifeature fusion layer to connect them. This multifeature fusion learning method can achieve good performance [13].

The topology-based 3D model descriptors analyze the topological structure of the 3D model to extract the topological connections and structural relations among different components. At present, this type of methods mainly includes attribute adjacency graph (AAG) [14], feature dependency graph (FDG) [15], skeleton graph [16], and Reeb graph [17, 18]. At present, the trend is to combine the topological structure of the 3D model and multiple views. For example, Su et al. propose multiview CNN (MVCNN), which takes multiview images of an object. This method has the potential strength of MVCNNs in sketch-based shape retrieval [19].

The descriptors based on projective views are the most promising because they transform 3D models into images, which allow image processing methods used for retrieval. In this type of descriptors, the light field descriptor (LFD) is the most popular because it is robust to transformations, noise, and model degeneracy [20]. In the LFD, a 3D model is projected to generate 100 binary images, which are rendered in different views for each model. This descriptor represents 3D models better than other descriptors, but its time complexity is heavy because the image number used for matching is large. Recently, these methods which combine the projective views and deep learning have achieved good performance. In these methods, deep learning models are trained to extract features from the 2D views and make classification. For example, Johns et al. propose pairwise method to bring CNN to generic multiview recognition, by first decomposing an image sequence into a set of image pairs, classifying each pair independently, and then learning an object classifier by weighting the contribution of each pair [21]. Ma et al. propose a method which extracted 2D Zernike moments from 2D projective views as the view saliency. Then the view saliency is used to boost a multiview CNN (VS-MVCNN) for 3D object recognition [22]. In the DeepPano, a panoramic view is used to represent the 3D model, and the CNN is designed to learn deep representations directly from the panoramic view [23]. The similar method is PANORAMA-NN [24], which also uses a panoramic view. In addition, Hegde and Zadeh use FusionNet to combine the representation of 2D projective views and the representation of model volume to learn new features, which yields a significantly better classifier than using either of the representations in isolation [25]. Qi et al. make a comprehensive study on the voxel-based CNNs and multiview CNNs for 3D object classification [26]. Elhoseiny et al. explore CNN architectures for combining object classification and pose estimation learned with multiview images, and this method takes a single image as input for its prediction [27]. Kanezaki et al. improve this method by aggregating predictions from multiple images captured from different viewpoints [28].

We can see that many methods have been effectively applied to 3D model recognition. However, there are several problems that need to be solved. First, current methods do not consider the similarity of 2D views when representing 3D models as 2D views. If cameras around 3D models are sparse, projective views cannot fully describe 3D models. If cameras are dense, redundant views will be generated, resulting in heavy time and space complexity. Second, a fixed number of projective views are used for similarity matching, which also leads to high computational complexity. To solve the above problems, we propose a novel 3D model retrieval method, which is improved in both index building and model retrieval. In the index building, 3D models in library are first converted into 2D projective views using the proposed projection method. Then representative views are selected from these 2D projective views by the proposed method based on the K-means. This method can reduce redundant views and improve the retrieval accuracy and efficiency. After that, the representative views are input into the learned CNN to extract features, which are organized as indexes by their labels. In retrieval, the input 3D model is first processed by the same way as that used in the index building to obtain representative views. Then all representative views of a model are classified into one category by the CNN and voting algorithm, and then only the features of one category rather than all categories are chosen to make similarity matching with these representative features. In

addition, we propose a novel similarity matching method, in which the number of views for retrieval is gradually increased until the evidence is enough to determine a 3D model. Therefore, model retrieval efficiency is improved substantially.

2. The Proposed Methodology

2.1. The Overall Scheme. As shown in Figure 1, the whole process of the proposed method can be divided into three steps: (1) 3D model representation and CNN training; (2) 2D representative view extraction and index building; (3) model retrieval. In the first step, 3D models are first converted into 2D projective views, and then these 2D projective views are used to train the CNN. In this part, a projection method is proposed to generate views. In the second step, 3D models are first converted into 2D projective views using the same projection method as that used in training. These views are then selected by the proposed method based on the *K*-means. The views which are closest to the centers of their own categories are selected as the representative views. Finally, these representative views are input into the learned CNN for feature extraction and index building. In the third step, the input can be an image or a 3D model. If the input is an image, the classification and retrieval are carried out directly. If the input is a 3D model, representative views are generated first by our projection method and representative view selection method, and then the representative views are input into learned CNN for classification and feature extraction. All representative views of the 3D model can be classified into the same category through the voting algorithm. Finally, the result model is found through the variable view matching method.

2.2. 3D Model Representation and CNN Training. Nowadays, CNNs have been used widely for object detection, scene recognition, texture recognition, and fine-grained classification. The CNN is also used in the proposed method because the CNN outperforms other methods in our task, and the views projected from 3D models can be large enough to learn a good CNN.

2.2.1. Multiview Representation of 3D Model. It is a key step to represent 3D models into 2D projective views. The main two factors in obtaining the projective views are the selection of projection method and rendering mode. Through experiments, we adopt the projection method based on region division and rendering method based on multilight sources. The steps are described as follows:

(1) Model preprocessing: the purpose of model preprocessing is to normalize the 3D model by limiting it to the unit sphere. First, the maximum and minimum values in the three coordinate directions are obtained by collecting the boundary information of the model and traversing the coordinates of all points. Then, the scaling and the position center of the model are calculated. Finally, the model is translated and scaled. The model preprocessing is shown in Figure 2.

(2) Selection of projective points: cameras are deployed on the sphere centered on the center of the 3D model. The spherical surface is divided into four uniform regions, with one camera deployed at the center of each region. Any other cameras are located in the bisectors which pass through the center. The angle between the bisectors is equal. The cameras placed on the bisectors are located in the middle points between the center points and the boundaries. The lens of each camera should point to the sphere center. The placement of the camera in each region is shown in Figure 3.

(3) Model rendering: in order to increase the information quantity contained in projective views and reduce the negative impact from the shadow of the model, we adopt the Phong Lighting Model [29] to render the model. Firstly, an ambient light of low intensity is used, and then six fixed weak light sources at the points (0, 0, 1), (0, 0, −1), (0, 1, 0), (0, −1, 0), (1, 0, 0), and (−1, 0, 0) are deployed. At last, a brighter point source is set at the position of each camera, which is turned on when views are acquired. The six weak light sources and their locations are shown in Figure 4.

In the proposed method, 40 projective views are used. A comparison of the proposed method and the LFD is shown in Figure 5, where the view generated by our method is shown in Figure 5(a) and that by the LFD is shown in Figure 5(b). We can see that the projective view obtained by our method is a grayscale image with information entropy of 0.462. In contrast, the projective view of the LFD is a binary image with information entropy of 0.287. Therefore, our method contains more detailed features.

2.2.2. CNN Training. In recent years, CNNs are widely used for image classification. At present, there are a lot of CNNs, such as the VGG, GoogleNet, ResNet, and DenseNet. It is reported that the ResNet can achieve the good performance on ImageNet. The ResNet adopts a unique "shortcut connection" which can effectively avoid gradient disappearance and ensure the training accuracy [30]. In our experiments, the ResNet50 achieved better performance than other types of deep neural networks, so it was used for feature extraction and classification. The ResNet50 consists of 49 convolutional layers and one fully connected layer. The structure of the ResNet50 is shown in Table 1.

2.3. Index Building. It is very important to build indexes for improving the efficiency of model retrieval. In this section, representative view selection is presented first, and then the index building based on the CNN is introduced.

2.3.1. Representative View Selection Based on K-Means. The 2D view number and the projection angle have an impact on the representation of 3D models. In current

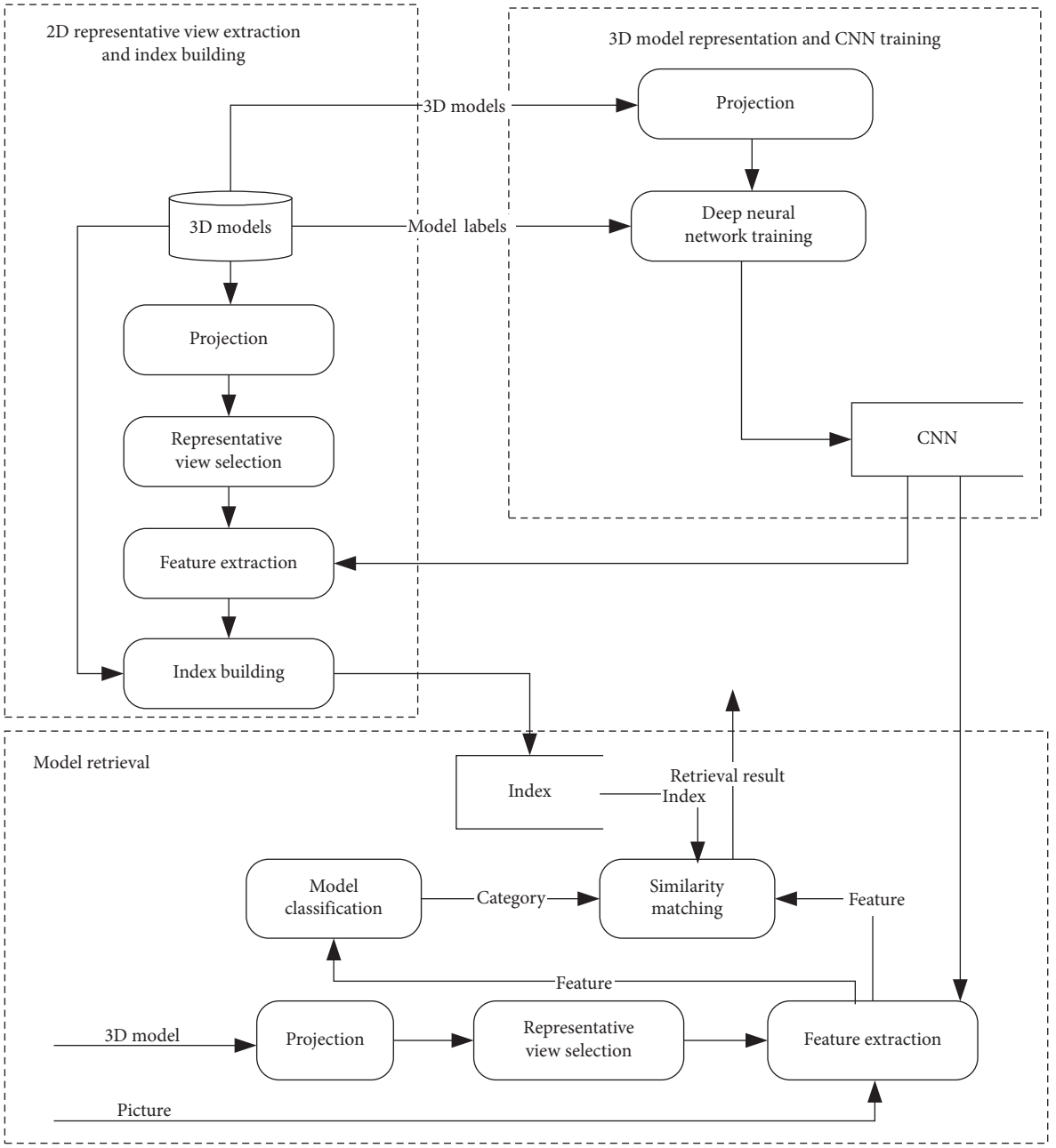


Figure 1: Retrieval process.

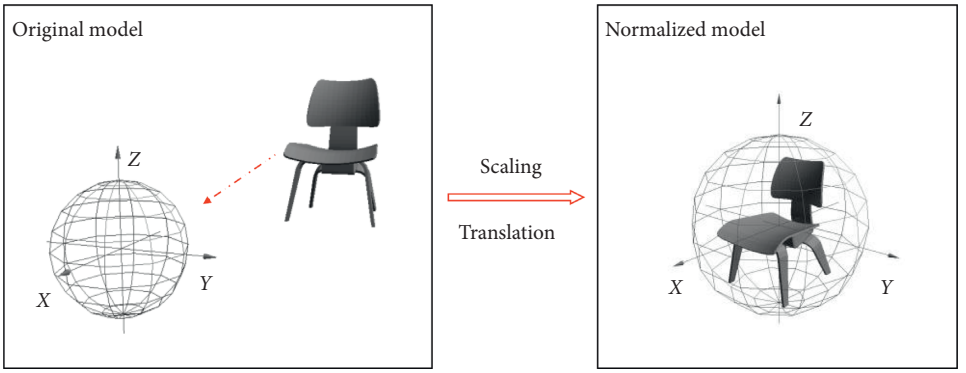
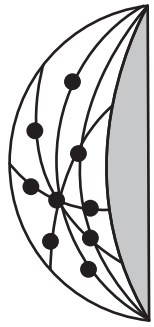
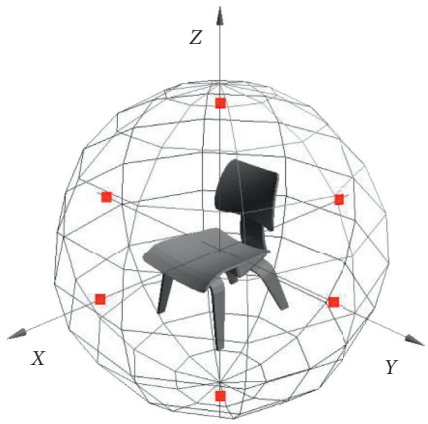


Figure 2: Model preprocessing.



Camera

Figure 3: Placements of the cameras.



Six fixed weak light sources

Figure 4: Placements of six fixed weak light sources.

methods, a large number of cameras are evenly distributed on the surface of the unit sphere to obtain 2D views. This way does not take the differences in model surface complexity into account. In fact, the part of the 3D model with large surface complexity needs more views to represent, while the part with small surface complexity can be well represented with fewer views. The 2D views projected by current methods have a large number of similar views, which cause amounts of redundancy. Therefore, it is necessary to keep only one view from similar views to make views more representative. In this paper, we propose a method to extract 2D representative views. In this method, the K-means is adopted to classify views into different categories according to their similarity, and then one representative view is chosen from each category. In this way, different 3D models may yield different numbers of 3D views.

As an unsupervised classification method, clustering classifies datasets without labels into several clusters [31]. One widely used algorithm for clustering is the K-means [32]. Its advantages are simplicity and local minimum convergence properties. However, it has a shortcoming that the number of clusters should be set manually. For each 3D model, the proposed method based on the K-means is implemented as follows:

Step 1: convert the 3D model into 40 2D projective views by the projection method proposed in Section 2.2.1

Step 2: cluster these 2D projective views using the K-means

Step 3: select the views which are closest to the centers of their own categories as the representative views

When the 2D views are clustered by the K-means, the number of categories K must be determined first. According to the experiment, 10–20 views can obtain good performance. Therefore, K is roughly set as 10–20, and then the elbow [33] method is used to determine the final value of K. If the 2D views of a 3D model are divided into K categories, K 2D representative views are obtained for the representation of a 3D model.

2.3.2. Index Building Based on CNN. The indexes of 3D models are built by inputting the 2D representative views into the ResNet50 and then organizing the output features according to their categories. As for input model Model_i, its representative views $W_{i1}, W_{i2}, \ldots, W_{in}$ are first generated. Then, these representative views are input into the learned ResNet50. The outputs of the 49th layer of ResNet50 are features of these representative views, denoted by $F_{i1}, F_{i2}, \ldots, F_{in}$. The outputs of the 50th layer of ResNet50 are the labels of these representative views. In this method, the task of 3D model classification is transformed into the classification of views. The index building process is shown in Figure 6.

2.4. Model Retrieval. The task of similarity matching is to find the most similar 3D model in the dataset according to the input. The input can be an image or a 3D model. If the input is an image, the features are directly extracted and the category is determined through the learned CNN. In a category, the output 3D model is found via the following equation:

$$i = \arg\min_{i,j} \text{dis}\left(W, F_{ij}\right), \tag{1}$$

where dis() is the function to compute the Euclidean distance, W is the features of the input image, F_{ij} is the features of jth view of the ith model, $1 \le i \le m$, m is the number of models in a category, $1 \le j \le n_i$, and n_i is the number of representative views of the ith model. The model i is the output result.

If the input is a 3D model, the model retrieval is realized in three steps: (1) generating 2D representative views; (2) inputting these views into CNN for feature extraction and classification. All representative views of a model may not be classified into the same category because of misclassification, so we adopt voting algorithm to determine one category for views of a model; (3) performing similarity matching. In order to improve the matching efficiency, we propose a similarity matching method which uses variable view numbers.

Let Category_Vector denote category vector with the cth element indicating the number of views classified into the cth category. Category_Vector is initialized as follows:

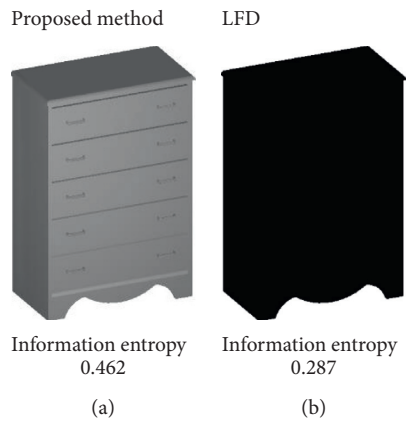


Figure 5: A comparison of two projection methods.

Table 1: The structure of ResNet50.

Layer name	Output size	50-layer
Conv1	112×112	7×7, 64, stride 2
	56×56	3×3 max pool, stride 2
Conv2_x	56×56	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$
Conv3_x	28×28	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$
Conv4_x	14×14	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 6$
Conv5_x	7×7	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$
	1×1	Average pool, 1000-d fc, softmax
FLOPs		3.8×10^9

$$\text{Category\_Vector} = [0, 0, \ldots, 0], \tag{2}$$

where Category_Vector is a c-dimensional vector corresponding c categories in a model library. When a representative view is assigned to the cth category, this vector is updated by

$$\text{Category\_Vector}[c] = \text{Category\_Vector}[c] + 1. \tag{3}$$

Finally, the category of the model is determined by

$$c = \arg\max_{c} \text{Category\_Vector}[c]. \tag{4}$$

After classification, the retrieval procedure is summarized in Algorithm 1. In order to improve retrieval efficiency, we design a flexible retrieval strategy: (1) if the distance between an input view and a view of a model in the library is small enough, i.e., dis $< \eta$, we can make sure that this model is what we need (output model); (2) if there are $C_{\text{threshold}}$ representative views belong to the same model in the same category, we can make sure that this model is what we need (output model); (3) if representative views are matched with different models of the same category, the cumulative distance value is calculated. If the cumulative distance value of a model is the minimum, the model is the output model.

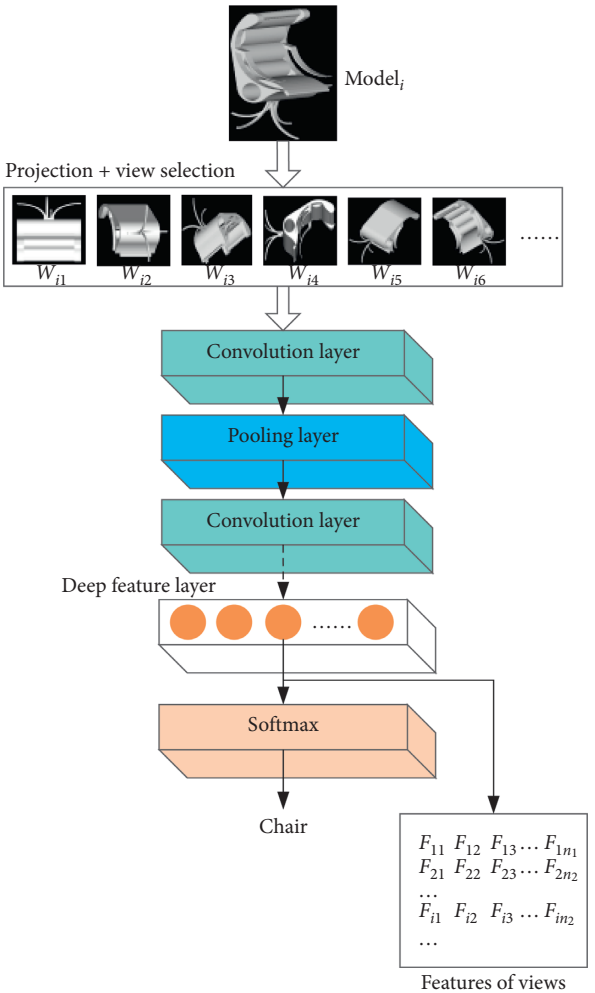


Figure 6: Index building process.

3. Experiments and Results

The experiments are conducted on an Intel i5 8400 + GTX 1060 PC. The proposed method is implemented based on the MXNET framework. The ResNet50 is used to build model indexes and implement model classification. The proposed

method is evaluated on the following two aspects: model classification and model retrieval.

3.1. Model Classification Evaluation. In this section, we compare the proposed method with the state-of-the-art methods. The evaluation is made on the following 3D model databases: McGill 3D Shape Benchmark [34] (a nonrigid 3D model dataset) and ModelNet10 and ModelNet40 [35] (two rigid 3D model datasets). Table 2 shows the detail information of these datasets.

We follow the training and testing splitting included in ModelNet10 and ModelNet40. ModelNet10 consists of 4899 models in 10 categories, and 3991 are used as the training dataset and 908 models are used as the test dataset. ModelNet40 consists of 12311 models in 40 categories, and 9843 are used as the training dataset and 2468 models are used as the test dataset. In the McGill, there are 255 models. 179 3D models are randomly selected for training and the remaining 76 3D models as the test dataset.

The model pretrained with the data of ImageNet is used as initialization parameters of the ResNet50. The learning rate is set as 0.01. The batch_size is set as 32 according to GPU size and training efficiency. In order to make the loss function converge quickly, the epoch is set as 200.

3.1.1. Representative View Selection. In the proposed projection method, each 3D model is presented as 40 views. In order to improve the efficiency of classification and retrieval, representative views are selected from the 40 projective views by the method proposed in Section 2.3.1. The number of representative views K has a great influence on the classification accuracy. In experiments, K is set as 5, 10, 20, and 30, respectively. Misclassified models of the proposed method given different K are shown in Table 3.

In the McGill, whatever K is, there is no misclassified model. The number of misclassified models in ModelNet10 and ModelNet40 decreases as K becomes larger. When K is 5, the number of misclassified models is the largest. When K is more than 20, the number of misclassified models is decreasing slowly. According to this result, we set the range of K as [10, 20].

The performance of the proposed method under different datasets and different conditions is shown in Table 4. Taking ModelNet10 as an example, there are 908 models in its training set, and each model has 40 2D views before representative view selection. Then the number of 2D views is 36320 (908×40). After representative view selection, each model has about 14 2D views, so the number of 2D views is 12742. The classification accuracy remains the same before and after the representative view selection method is used.

We can see from Table 4 that our representative view selection method does not cause performance degradation on McGill and ModelNet10. The classification accuracy on ModelNet40 only decreases by 0.9% after our representative view selection. It should be noted that our representative view selection can significantly reduce the number of views to about 1/3. A smaller number of views lead to higher efficiency of the 3D model classification and retrieval. The experiment in the following section adopts representative views for model classification and retrieval. For each model, about 14 projective views are enough to obtain a good performance.

3.1.2. Comparison of Classification Algorithms Based on Views. We compare the proposed method with several traditional methods, and the results are shown in Table 5. We can see that our proposed method has achieved the best performance in ModelNet10, with a recognition accuracy of 94.10%. In addition, it has achieved a recognition accuracy of 92% in ModelNet40, which is just 0.9% lower than that of VS-MVCNN. Although VS-MVCNN outperforms the proposed method, it needs 80 views.

Our proposed method can achieve 100% recognition accuracy in McGill (shown in Figure 7). This indicates that the proposed method performs well on both rigid and nonrigid 3D datasets.

3.1.3. Classification Result Analysis. The confusion matrix of the proposed method in ModelNet10 is shown in Figure 8. We can see that the proposed method can achieve an accuracy of 100% in classes of bed, chair, and monitor, an accuracy of more than 90% in the classes of bathtub, desk, sofa, and toilet, and an accuracy of less than 90% in classes of dresser, night_stand, and table (respectively, 88%, 84%, and 83%). The accuracy in table class is the worst, with 15% of models being misclassified as desk class and 2% of models being misclassified as night_stand class. The reason is that the models in table class and the models in desk class are extremely similar to each other.

We can see from Figure 9 that the models in the dresser class and night_stand class are extremely similar, which leads to misclassification. The misclassification of these models does not matter for users because the two models are either the same or similar enough.

The advantage of our method is that it can obtain high accuracy given a small number of views. Especially on McGill, the recognition accuracy is 100%. The reason is that there are great differences between the classes on McGill, and multiple views can better represent 3D models from different angles, leading to superior performance. However, on ModelNet10 and ModelNet40, the proposed method does not have good performance on some classes, such as the table class and desk class, or night_stand class and dresser class. The reason is that there is no obvious difference between the classes of ModelNet10, as well as ModelNet40. It is easy to make mistake for any classification method.

3.2. Retrieval Experiment. Our retrieval method is based on the classification results. The input is classified before similarity matching. The advantage is that similarity is calculated between the input and the models in one category rather than all categories, so it can greatly reduce the searching scope and computation complexity. In the following section, the similarity matching method is evaluated and analyzed on the rigid datasets and the nonrigid dataset, respectively.

input: W_l is the features of representative views of input model, $l = \{1, 2, 3, \ldots, p\}$,
F_{ij} is the features of jth view of the ith model in dataset,
m is the number of models in a category,
n_i is the number of representative views of the ith model,
η is the minimum distance,
Distance_Vector is the distance vector, Distance_Vector = $[0, 0, \ldots, 0]$,
Count is the counting vector, it is used to record the number of views that are classified into each category, Count = $[0, 0, \ldots, 0]$
output: i_{searched},
$i_{\text{searched}} = -1$;
$\eta = 1.5$;
for $(1 \le l \le 14)$
{
$k_{\min} = 0$;
$\text{dis}_{\min} = 1000000$;
for $(1 \le i \le m)$
{
$\text{dis} = \min_j \text{dis}(W_l, F_{ij})$ $(j = 1, 2, \ldots, \text{dis} < \eta)$;
if $(\text{dis} < \eta)$ $\{i_{\text{searched}} = i$; return; $\}$
Distance_Vector$[i]$ = Distance_Vector$[i]$ + dis;
if $(\text{dis} < \text{dis}_{\min})\{k_{\min} = i$; $\text{dis}_{\min} = \text{dis}$;
}
}
Count$(k_{\min})$ = Count$(k_{\min})$ + 1;
for $(1 \le i \le m)$
{
if (Count(i) = $C_{\text{threshold}}$) $\{i_{\text{searched}} = i$; return; $\}$
}
}
if $(i_{\text{searched}} = -1)$ $i = \arg\min_i$ Distance_Vector$[i]$;
return i_{searched};

Algorithm 1: Similarity matching algorithm.

Table 2: 3D model datasets.

3D model dataset	Models	Classes
McGill	255	10
ModelNet10	4899	10
ModelNet40	12311	40

Table 3: Misclassified models given different K.

K	5	10	20	30
McGill	0	0	0	0
ModelNet10	77	62	56	55
ModelNet40	212	191	182	183

Table 4: Views and classification accuracy (%).

	McGill		ModelNet10		ModelNet40	
	Before	After	Before	After	Before	After
Views	960	362	36320	12742	98720	34526
Accuracy	100	100	94.10	94.10	92.90	92.0

3.2.1. Retrieval Experiment for Rigid Datasets. Our shape descriptors are compared against the spherical harmonics descriptor (SPH) [10], LFD [20], 3D ShapeNets [6], DeepPano [23], PANORAMA-NN [24], View Inter-Prediction GAN (VIPGAN) [36] and Ma et al.'s method [37]. The result of the mean average precision (MAP) is shown in Table 6. We can see that the MAP of our proposed method is obviously higher than those of other methods. There are two reasons for this: (1) classification is made before retrieval because the accuracy of the proposed classification method is high enough to ensure the good retrieval accuracy, and (2) the voting mechanism is adopted. Some views of an input model are easily misclassified due to their high similarity. Through voting mechanisms, these misclassified views can be reclassified correctly.

Table 5: Classification accuracy compared with other methods (%).

Algorithm	Views	ModelNet10	ModelNet40
DeepPano [23]	1	88.66	82.54
PANORAMA-NN [24]	1	91.10	90.70
Pairwise [21]	12	93.20	91.10
FusionNet [25]	60	93.11	90.80
VS-MVCNN [22]	80	93.50	92.90
Ours	14	94.10	92.00

The precision-recall curves are shown in Figures 10 and 11. We can see that our method outperforms other state-of-the-art methods. The precision-recall curve of the proposed method is stable, while those of other methods gradually decrease with the increase of recall. Taking Figure 10 as an example, when the recall rate is less than 0.2, the PANORAMA-NN and Ma et al.'s

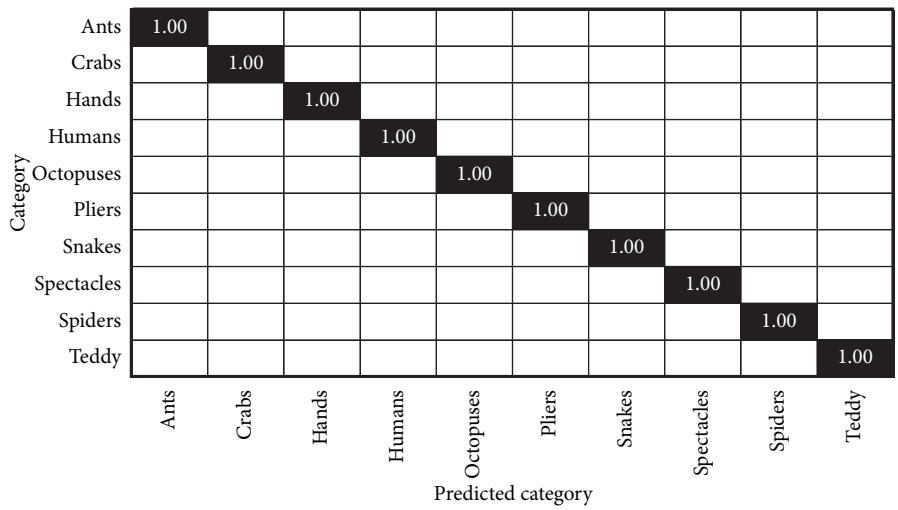


Figure 7: Classification results on McGill.

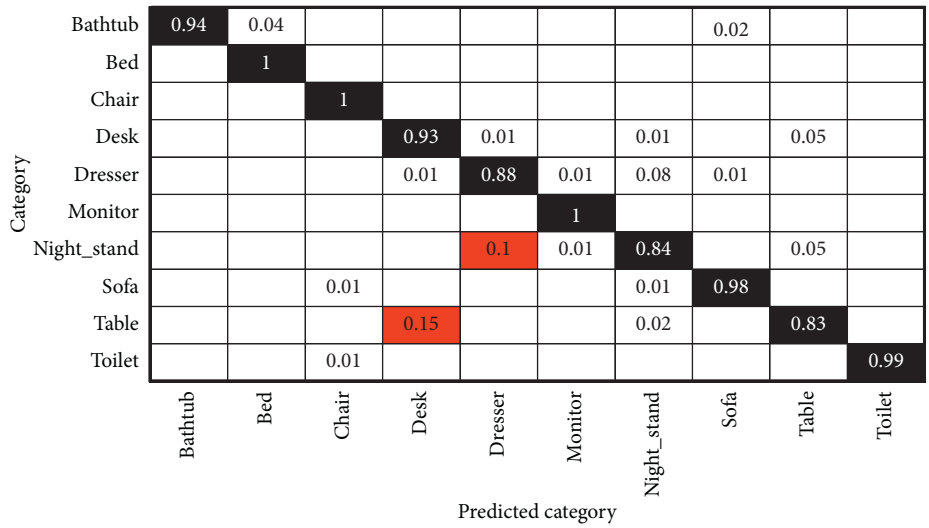


Figure 8: ModleNet10 classification results.

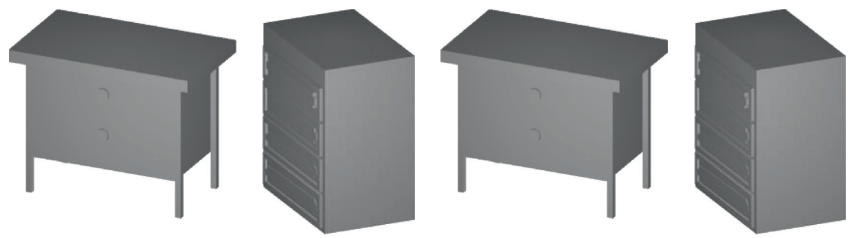


Figure 9: Similar models in different categories.

method perform better than our method. However, when the recall rate is larger than 0.9, the precision rates of the two methods decrease rapidly. In particular, the precision rate of Ma et al.'s method decreases to 0.1 when the recall rate is close to 1. The precision-recall curves of the DeepPano and VIPGAN are similar to that of the proposed method when the recall rate is less than 0.9. However, their precision rates decrease rapidly when the recall rate is close to 1. The SPH performs the worst.

Table 6: The comparison of the proposed method and other methods (MAP, %).

Algorithm	Dataset	
	ModelNet10	ModelNet40
SPH [10]	45.9	34.4
LFD [20]	49.8	40.9
3D ShapeNets [6]	69.2	59.9
DeepPano [23]	84.2	76.8
PANORAMA-NN [24]	87.4	83.5
VIPGAN [36]	90.6	89.2
Ma et al. [37]	93.1	84.3
Ours	94.1	92.0

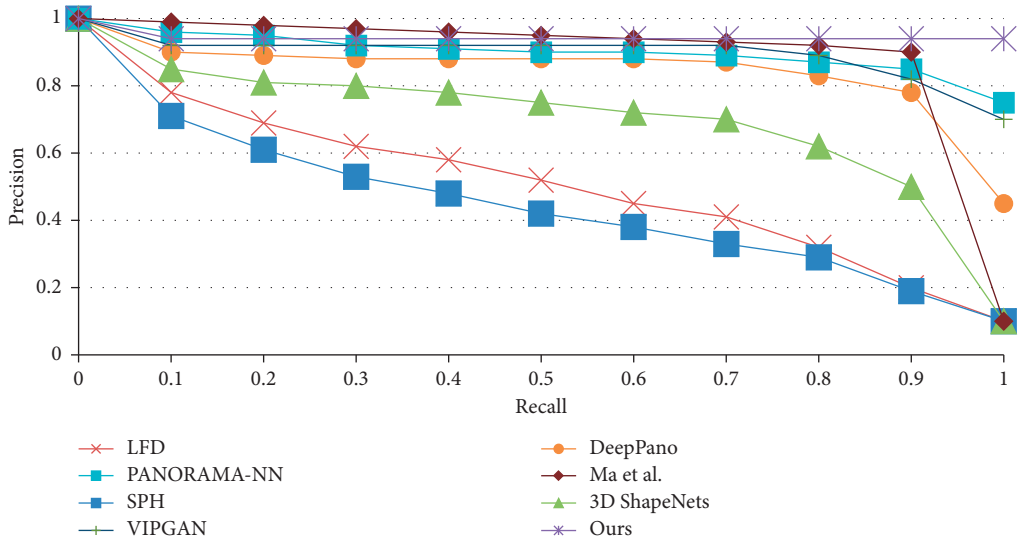


Figure 10: The comparison of precision-recall curves for various methods on ModelNet10.

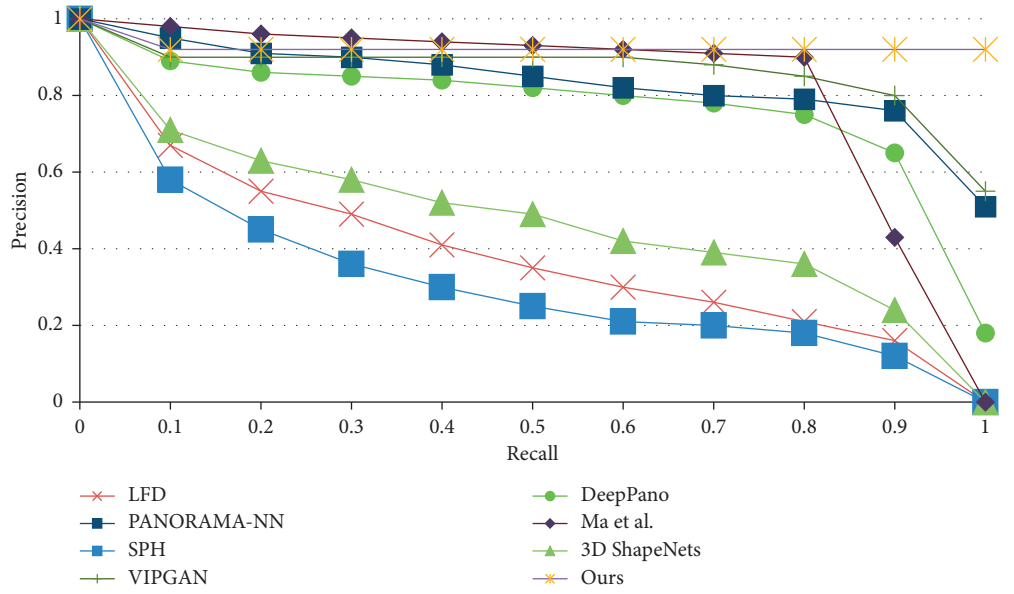


Figure 11: The comparison of precision-recall curves for various methods on ModelNet40.

The LFD is slightly better than the SPH. The 3D ShapeNets is in the middle of these eight methods. The precision rates of these three methods decrease from 1 to 0 with the increase in recall.

3.2.2. Retrieval Experiment for Nonrigid Dataset. The used nonrigid dataset is McGill. We compare our proposed method to the heat kernel signature (HKS) [11], the wave kernel signature (WKS) [12], the CBoFHKS [38], the discriminative autoencoder-based shape descriptor (DASD) [39], the multifeature fusion learning (MFFL) [13], and the learning-based multiple pooling fusion (LMPF) [40]. Table 7 shows the retrieval results measured by the Nearest Neighbor (NN), First Tier (FT), Second Tier (ST), and Discounted Cumulative Gain (DCG).

We can see from Table 7 that our proposed method achieves the best performance on the NN, FT, ST, and DCG measures. And the performance of the proposed method on the nonrigid dataset is better than that on the rigid dataset. The reason is that we use the well-trained CNN to classify the models in McGill. The classification accuracy is 100%, so the retrieval accuracy is also 100%. In summary, our method obtains good performance on both rigid and nonrigid datasets.

3.2.3. Retrieval Efficiency Analysis. Experiments show that similarity matching consumes the most time during 3D model retrieval. Taking ModelNet10 as an example, there are 908 models in the test set and 3991 models in the training set. Each model has 40 views, so the test set contains 36320 views and the training set contains 159640 views. If all views are used for similarity matching, the time complexity is large. Table 8 shows the comparison of the number of views before and after representative view selection in ModelNet10.

We can see that the view number in the test set decreases from 36320 to 12742 and the view number in the training set decreases from 159640 to 56613 through representative view selection. The view number is reduced by 2/3 after representative view selection, so this method can effectively reduce redundant views and greatly improve the retrieval efficiency.

In ModelNet10, the training set consists of 3991 models, and these models are divided into 10 classes, with each class consisting of 399 models on average. After applying representative view selection, the number of similarity matching is reduced from 638400 ($40 \times 399 \times 40$) to 78204 ($14 \times 399 \times 14$) (reduced by 87.8%).

The variable view matching method can further improve the matching efficiency. In this paper, η is defined as the similarity of two views generated by two adjacent projective points of the same model. We call η as adjacent view distance. The smaller η is, the higher the accuracy is. We take ModelNet10 as an example to analyze η under our projection method. Adjacent projection points are shown in Figure 12.

Experiments show that the adjacent view distances of any two views are different. In the same category, the minimum adjacent view distance is chosen as the representative to form the list of adjacent view distance. Table 9 shows the average adjacent view distances when different numbers of models are selected for each category. Taking the bathtub class as an example, when the model number is 1, the minimum adjacent view distance is 1.705. When the model number is 20, the average adjacent view distance is 1.995. The adjacent view distance of the table class is the smallest and that of the bed class is the largest. The reason is that the model complexity is different. The models in the table class are simple, while the models in the bed class are more complex than others. The last row of Table 9 shows the average adjacent view distance of all categories with model numbers of 20, 10, 5, and 1. We can see that when the number of models is 1, the average adjacent view distance is the smallest with 1.6418. When the number of models is 20, the average adjacent view distance is the biggest at 1.8572. In order to improve the efficiency and accuracy of 3D model retrieval, η is set as 1.5.

In Algorithm 1, there are three conditions to finish similarity matching. The view numbers used under the three conditions are 1, 5, and 14, respectively, i.e., $C_{\text{threshold}}$ is set as 5. The results on ModelNet10 are shown in Table 10, where η is 1.5. For example, in bathtub, there are 3 models under condition 1. That is to say, these 3 models can be retrieved by only using one view. And there are 4 models under condition 2 and 43 models under condition 3. If we do not use the variable view matching, all models are retrieved by using 14

Table 7: Performance comparison on McGill.

Method	NN	FT	ST	DCG
HKS [11]	0.8190	0.6220	0.7440	0.8270
WKS [12]	0.9140	0.7750	0.8660	0.9140
CBoFHKS [38]	0.9010	0.7780	0.8760	0.8910
DASD [39]	0.9880	0.7820	0.8340	0.9550
MFFL [13]	0.9710	0.9050	0.9810	0.9630
LMPF [40]	0.9810	0.8610	0.9594	0.9579
Ours	1.0000	1.0000	1.0000	1.0000

Table 8: The number of views before and after representative view selection.

Dataset	Before		After	
	Total	Average	Total	Average
Test set	36320	40	12742	14
Training set	159640	40	56613	14

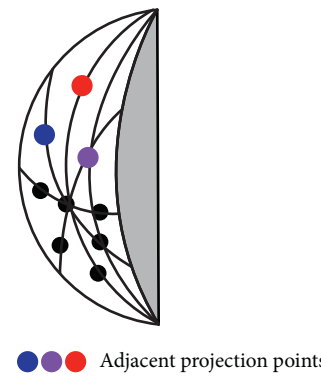


Figure 12: Adjacent projection points.

Table 9: The average adjacent view distance.

Category	Model number			
	20	10	5	1
Bathtub	1.995	1.897	1.774	1.705
Bed	2.304	2.259	2.221	2.186
Chair	1.627	1.546	1.430	1.351
Desk	2.142	2.052	1.980	1.900
Dresser	2.145	2.091	2.050	1.986
Monitor	1.700	1.666	1.637	1.581
Night_stand	1.995	1.920	1.857	1.741
Sofa	1.948	1.900	1.859	1.796
Table	0.751	0.531	0.485	0.441
Toilet	1.965	1.895	1.820	1.731
Average	1.857	1.775	1.7113	1.641

Table 10: The model numbers under different conditions.

Category	Condition 1 (1 view)	Condition 2 (5 views)	Condition 3 (14 views)	Traditional method (14 views)
Bathtub	3	4	43	50
Bed	10	3	87	100
Chair	18	16	66	100
Desk	3	3	80	86
Dresser	12	10	64	86
Monitor	8	4	88	100
Night_stand	10	15	61	86
Sofa	11	5	84	100
Table	4	9	87	100
Toilet	53	12	35	100
Models	132	81	695	908
Views	132	405	9730	12712

views. In ModelNet10, if we use variable view matching, the number of all views is 10267 (132 + 405 + 9730), while that of the traditional method is 12742. The number of views is reduced by 2475. That is to say, the average number of views for retrieval of each model is reduced to 11. Through variable view matching, the average number of similarity matching of each model is approximately 61446 ($11 \times 399 \times 14$). Compared with only using representative view selection method, the number of similarity matching is further reduced by 21.4%.

4. Conclusion

With the increase of 3D models, the degradation of retrieval accuracy and efficiency becomes a serious problem for 3D model retrieval systems. An efficient 3D model retrieval method is proposed in this paper. The efficiency of the proposed method is improved in three aspects: (1) Efficient indexes are built through the representative view selection and the feature extraction with the CNN. And then features are organized via their labels. In this way, the 3D models are represented more efficient and the number of used views is reduced substantially. (2) The number of similarity matching is reduced by classification before retrieval. In retrieval, 2D views of the input model are classified into one category with the CNN and voting mechanism, and then, only the features of this category, rather than all categories, are chosen to make similarity matching. (3) Variable view matching method is proposed. The retrieval of some models can be terminated ahead of time. The accuracy of our proposed method is improved in two aspects: (1) The classification of input models is made before retrieval. Our classification method obtains good performance, so the retrieval accuracy and efficiency are guaranteed. (2) The voting mechanism is used to classify input 3D models. Through the voting mechanisms, the misclassified views can be reclassified correctly.

Although the proposed 3D model retrieval method demonstrates great improvement in both accuracy and efficiency, similar 3D models are easy to be misclassified. Therefore, we will study how to improve the discrimination of model representation in our future work.

Data Availability

Previously reported ModelNet10 and ModelNet40 data are used to support this study and are available at http://modelnet.cs.princeton.edu/. These prior studies (and datasets) are cited at relevant places within the text as reference [20]. The McGill 3D Shape Benchmark data are used to support this study and are available at http://www.cim.mcgill.ca/~shape/benchMark/. These prior studies (and datasets) are cited at relevant places within the text as reference [19]. We also called it McGill and McGill10 in our paper.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This research was partly supported by the National Natural Science Foundation of China (61673142), the Natural Science Foundation of Heilongjiang Province of China (JJ2019JQ0013), the Outstanding Youth Talent Foundation of Harbin of China (2017RAYXJ013), the University Nursing Program for Young Scholars with Creative Talents in Heilongjiang Province (UNPYSCT-2016034), and the Foundation of Education Department of Heilongjiang Province (12511096).

References

[1] P. Pal and K. K. Ghosh, "Estimating digitization efforts of complex product realization processes," *The International Journal of Advanced Manufacturing Technology*, vol. 95, no. 9–12, pp. 3717–3730, 2018.

[2] A. Zeng, S. Song, M. Niessner, M. Fisher, J. Xiao, and T. Funkhouser, "3DMatch: learning local geometric descriptors from RGB-D reconstructions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1802–1811, Honolulu, HI, USA, July 2017.

[3] M.-D. Ahrend, H. Noser, R. Shanmugam et al., "Development of generic Asian pelvic bone models using CT-based 3D

statistical modelling," *Journal of Orthopaedic Translation*, vol. 20, pp. 100–106, 2020.

[4] S. Tao, Z. Huang, L. Ma, S. Guo, S. Wang, and Y. Xie, "Partial retrieval of CAD models based on local surface region decomposition," *Computer-Aided Design*, vol. 45, no. 11, pp. 1239–1252, 2013.

[5] Z. Han, M. Shang, Z. Liu et al., "SeqViews2SeqLabels: learning 3D global features via aggregating sequential views by RNN with attention," *IEEE Transactions on Image Processing*, vol. 28, no. 2, pp. 658–672, 2019.

[6] Z. R. Wu, S. R. Song, A. Khosla et al., "3D ShapeNets: a deep representation for volumetric shapes," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1912–1920, IEEE Computer Society Press, Boston, MA, USA, June 2015.

[7] D. Maturana and S. Scherer, "Voxnet: a 3D convolutional neural network for real-time object recognition," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 922–928, Hamburg, Germany, September 2015.

[8] Y. Li, S. Pirk, H. Su et al., "FPNN: field probing neural networks for 3D data," in *Proceedings of the Advances in Neural Information Processing Systems (NIPS)*, pp. 307–315, Barcelona, Spain, December 2016.

[9] J. Wu, C. Zhang, and T. Xue, "Learning a probabilistic latent space of object shapes via 3D generative-adversarial modeling," in *Proceedings of the Advances in Neural Information Processing Systems (NIPS)*, pp. 82–90, Barcelona, Spain, December 2016.

[10] M. Kazhdan, T. Funkhouser, and S. Rusinkiewicz, "Rotation invariant spherical harmonic representation of 3D shape descriptors," in *Proceedings of the Symposium on Geometry Processing*, vol. 6, pp. 156–164, Aachen, Germany, June 2003.

[11] J. Sun, M. Ovsjanikov, and L. Guibas, "A concise and provably informative multi-scale signature based on heat diffusion," *Computer Graphics Forum*, vol. 28, no. 5, pp. 1383–1392, 2009.

[12] M. Aubry, U. Schlickewei, and D. Cremers, "The wave kernel signature: a quantum mechanical approach to shape analysis," in *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pp. 1626–1633, Barcelona, Spain, November 2011.

[13] H. Zeng, Y. Liu, S. Li, J.-Y. Che, and X. Wang, "Convolutional neural network based multi-feature fusion for non-rigid 3D model retrieval," *Journal of Information Processing Systems*, vol. 14, no. 1, pp. 176–190, 2018.

[14] N. Bore, P. Jensfelt, and J. Folkesson, "Querying 3D data by adjacency graphs," in *Proceedings of the International Conference on Computer Vision Systems*, pp. 243–252, Lecture Notes in Computer Science, Copenhagen, Denmark, July 2015.

[15] B. Ding, Z. Zhang, X. Y. Yu, and Y.-B. He, "3D CAD model retrieval based on GA-ACO," in *Proceedings of the IFOST*, vol. 2, pp. 36–41, Ulaanbaatar, Mongolia, July 2013.

[16] H. Liu, J. Xia, J. Chen, and J. Wang, "Detection of hierarchical intrinsic symmetry structure in 3D models," *Computers & Graphics*, vol. 70, pp. 8–16, 2018.

[17] A. Liu, Z. Wang, W. Nie, and Y. Su, "Graph-based characteristic view set extraction and matching for 3D model retrieval," *Information Sciences*, vol. 320, pp. 429–442, 2015.

[18] N. Karmakar, A. Biswas, and P. Bhowmick, "Reeb graph based segmentation of articulated components of 3D digital objects," *Theoretical Computer Science*, vol. 624, pp. 25–40, 2016.

[19] H. Su, S. Maji, E. Kalogerakis, and E. Learned-Miller, "Multi-view convolutional neural networks for 3D shape recognition," in *Proceedings of the 2015 International Conference on Computer Vision*, pp. 945–953, Santiago, Chile, December 2015.

[20] D.-Y. Chen, X.-P. Tian, Y.-T. Shen, and M. Ouhyoung, "On visual similarity based 3D model retrieval," *Computer Graphics Forum*, vol. 22, no. 3, pp. 223–232, 2003.

[21] E. Johns, S. Leutenegger, and A. J. Davison, "Pairwise decomposition of image sequences for active multi-view recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3813–3822, IEEE Computer Society Press, Las Vegas, NV, USA, July 2016.

[22] Y. X. Ma, B. Zheng, Y. L. Guo et al., "Boosting multi-view convolutional neural networks for 3D object recognition via view saliency," in *Proceedings of the Chinese Conference on Image and Graphics Technologies*, Springer, Beijing, China, pp. 199–209, June 2017.

[23] B. Shi, S. Bai, Z. Zhou, and X. Bai, "DeepPano: deep panoramic representation for 3-D shape recognition," *IEEE Signal Processing Letters*, vol. 22, no. 12, pp. 2339–2343, 2015.

[24] K. Sfikas, T. Theoharis, and I. Pratikakis, "Exploiting the PANORAMA representation for convolutional neural network classification and retrieval," in *Proceedings of the 10th Eurographics Workshop on 3D Object Retrieval*, pp. 1–7, Lyon, France, April 2017.

[25] V. Hegde and R. Zadeh, "FusionNet: 3D object classification using multiple data representations," in *Proceedings of the Sixth International Conference on Learning Representations (ICLR)*, pp. 1–10, Vancouver, Canada, May 2018.

[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 *Proceedings of the 29th IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5648–5656, Las Vegas, NV, USA, June 2016.

[27] M. Elhoseiny, T. El-Gaaly, A. Bakry et al., "A comparative analysis and study of multiview CNN models for joint object categorization and pose estimation," in *Proceedings of the 33rd International Conference on Machine Learning (ICML)*, pp. 1402–1422, New York, NY, USA, June 2016.

[28] A. Kanezaki, Y. Matsushita, and Y. Nishida, "RotationNet: joint object categorization and pose estimation using multiviews from unsupervised viewpoints," in *Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5010–5019, Salt Lake City, UT, USA, June 2018.

[29] S. M. Woo, S. H. Lee, J. S. Yoo, and J.-O. Kim, "Improving color constancy in an ambient light environment using the Phong reflection model," *IEEE Transactions on Image Processing*, vol. 27, no. 4, pp. 1862–1877, 2017.

[30] K. He, X. Zhang, S. Ren et al., "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, Las Vegas, NV, USA, June 2016.

[31] J.-Y. Chen and H.-H. He, "A fast density-based data stream clustering algorithm with cluster centers self-determined for mixed data," *Information Sciences*, vol. 345, pp. 271–293, 2016.

[32] J. Xu, J. Han, and F. Nie, "Discriminatively embedded k-means for multi-view clustering," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5356–5364, Las Vegas, NV, USA, June 2016.

[33] P. Bholowalia and A. Kumar, "EBK-means: a clustering technique based on elbow method and k-means in WSN," *Computer Applications*, vol. 105, no. 9, pp. 17–24, 2014.

[34] K. Siddiqi, J. Zhang, D. Macrini, A. Shokoufandeh, S. Bouix, and S. Dickinson, "Retrieving articulated 3-D models using

medial surfaces," *Machine Vision and Applications*, vol. 19, no. 4, pp. 261–275, 2008.

[35] P. Shilane, P. Min, M. Kazhdan, and T. Funkhouser, "The princeton shape benchmark," in *Proceedings of the Shape Modeling Applications*, pp. 167–178, IEEE Computer Society Press, Genova, Italy, June 2004.

[36] Z. Han, M. Shang, Y.-S. Liu, and M. Zwicker, "View inter-prediction GAN: unsupervised representation learning for 3D shapes by learning global shape memories to support local view predictions," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 1, pp. 8376–8384, 2019.

[37] C. Ma, Y. Guo, J. Yang, and W. An, "Learning multi-view representation with LSTM for 3D shape recognition and retrieval," *IEEE Transactions on Multimedia*, vol. 21, no. 5, pp. 1169–1182, 2018.

[38] Z. Lian, A. Godil, T. Fabry et al., "SHREC'10 Track: non-rigid 3D shape retrieval," in *Proceedings of the Eurographics Workshop on 3D Object Retrieval*, pp. 107–120, Zurich, Switzerland, May 2015.

[39] J. Xie, G. Dai, F. Zhu, E. K. Wong, and Y. Fang, "Deep shape: deep-learned shape descriptor for 3D shape retrieval," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 7, pp. 1335–1345, 2017.

[40] H. Zeng, Q. Wang, L. Chen, and W. Song, "Learning-based multiple pooling fusion in multi-view convolutional neural network for 3D model classification and retrieval," *Journal of Information Processing Systems*, vol. 15, no. 5, pp. 1179–1191, 2019.