Retrieving 3D CAD Models Using 2D Images with Optimized Weights

Liang Li[1,2], Hanzi Wang, Tat-Jun Chin, David Suter
[1]School of Computer Science
The University of Adelaide
Adelaide, Australia

Shusheng Zhang
[2]School of Mechanical Engineering
Northwestern Polytechnical University
Xi'an, China

***Abstract*— An effective method for retrieving 3D models is to represent and discriminate them with their 2D images projected from multiple viewpoints. Such view-based methods conform more closely to human visual recognition for 3D model retrieval, since the human retina essentially captures 2D images. However, most of the existing view-based methods do not take into account that different views have different importance even though they belong to the same object. To address this problem, we propose a novel view-based method for 3D CAD model retrieval. First, the PHOG descriptor is employed to describe the 2D images projected from a model. Then, Lagrange multipliers, vector quantization and a Support Vector Machine (SVM) are used to adaptively assign an optimal weight to each projected image. The similarity between a 3D query model and a 3D object in database is determined by the likeness of their corresponding 2D images associated with optimal weights. The effectiveness of the proposed method is shown in the experimental part.**

Keywords-Content-based 3D model retrieval; PHOG; Lagrange mulitpliers; vector quantization; SVM

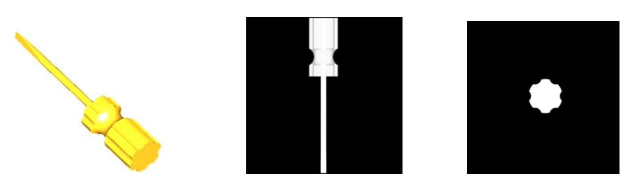
I. INTRODUCTION

The rapid advances in 3D modeling technique in past decades make 3D model acquisition increasingly simply and easy. Nowadays, 3D model are widely used in diverse domains, such as Engineering, Computer Vision and Computer Graphic. Its popularity raises interests in research on content-based 3D model retrieval techniques. In the domain of engineering, the reuse of 3D CAD models can be realized through the retrieval of the existing models, so as to effectively support and accelerate the process of product development and save the cost of design and manufacturing [1].

The current content-based 3D model retrieval techniques can be mainly classified into 4 categories based on feature representation: feature-vector-based methods, statistic-based methods, graph-based methods and view-based methods [1]. Among these methods, the view-based methods have been proven to give superior performance [2, 3]. The concept of the view-based methods comes from that two 3D models are similar if they look alike from every viewing angle. In view-based methods, a 3D model is represented by a set of 2D images projected from multiple viewpoints which are uniformly distributed on a viewing sphere, and the similarity of a query 3D model and a 3D object in database is calculated by aggregating the similarity of their corresponding projected images. Some global image feature descriptors (such as 2D shape distributions [4], Fourier transform and Zernike moments [5, 6]) are commonly used to describe the images. In recent years, local feature descriptors such as SIFT are employed in some works [7, 8] for their robustness towards noise sensitivity and redundancy.

The main disadvantage in most of the existing view-based retrieval methods is that all the 2D views from a 3D model are simply assumed to have the same importance. They ignore the reality that some 2D views may contain more information than those from the others [9]. For example, it can be seen apparently in Figure 1 that the front view of the screwdriver has more information than the top view. Therefore, it is necessary to differentiate the importance of different views. Some approaches are presented to enhance the view-based methods. In [9, 10], the authors employ the Bayesian approach to select more distinctive views after the projected images are generated. In [11], the views of the 3D models are associated with the corresponding weights which are acquired from the relevant feedback.

In this paper, we address the problem of 3D CAD model retrieval, and propose a new view-based retrieval method. Given a query model, our goal is to adaptively assign an optimal weight to each of its views and use the weighted views for retrieval. Our idea is inspired by the approach presented in [11], where an optimization algorithm is employed for improving retrieval performance. However, a different strategy is used in this paper. First, a local image feature descriptor called PHOG descriptor [12] is employed to construct a 3D model feature descriptor. Second, we use the vector quantization technique and a Support Vector Machine (SVM) to adaptively assign an optimal weight to each view of the query 3D model.



(a) (b) (c)

Figure 1. (a) is a screwdriver model; (b) and (c) are respectively the front view and top view

The rest of the paper is organized as follows: the overview of the proposed method is given in Section 2. The representation of 3D CAD model features and the weight optimization are described respectively in Section 3 and Section 4. Experimental results are provided in Section 5. Finally, we draw the conclusion in Section 6.

II. OVERVIEW OF THE METHOD

The framework of our method is illustrated in Figure 2. We start by carrying out feature extraction on all of the 3D CAD models in database. Each model is represented by 18 images projected from 18 viewpoints. A 3D feature descriptor is the concatenation of the PHOG descriptors associated with the 18 images. Afterwards, by using Lagrange multiplier algorithm, each model descriptor is associated with a weight vector. The elements of the vector are the optimal weights corresponding to the views of a model. Vector quantization is conducted on the weight vectors of the models in database. Those weight vectors are clustered and we represent each cluster by its centroid. Next, the models whose associated weight vectors are in the same cluster are classified into one group. Thus, once the group label of a model feature descriptor is known, the cluster label of its associated weight vector can be derived, and vice versa. A SVM is learned by using the feature descriptors and its associated group label.

When a query model is given, its feature is extracted in the same manner to the models in database. The group label of its feature descriptor, which is also the cluster label of its associated weight vector, is predicted by the SVM at first. Then, we acquire its corresponding optimal weight vector, which is the centroid of the weight vector cluster. Finally, the similarity between the query model and an object in database is calculated by comparing the similarity of their feature descriptors associated with the optimal weights.

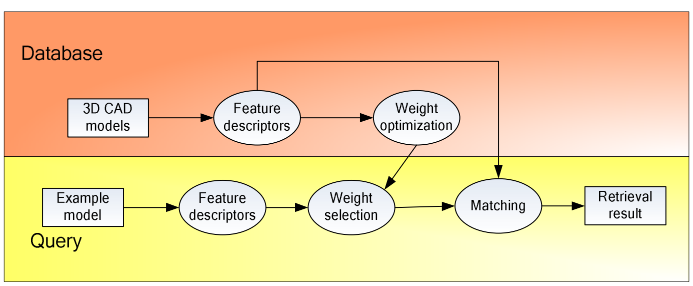


Figure 2. Framework of our method

III. FEATURE REPRESENTATION

It is noted [5] that a 3D model can be roughly represented by the 2D projected images from 15-20 viewpoints distributed evenly on the surface of the model. Accordingly, we use 18 images to describe a 3D model. The images are captured from the viewpoints placed at 18 vertices of the 32-hedron, as shown in Figure 2.

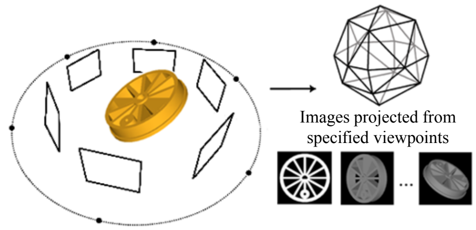


Figure 3. View projection

We use the PHOG descriptors to describe the images. This descriptor is capable of describing the spatial layout of local image shape by using a pyramid representation. Firstly, a Canny Edge detector is employed to find the edge contours of the shape. We then divide the image into 4^l areas according to the pyramid resolution level l. Next, the histogram of the edge orientations H will be computed for each area and be quantized into r bins, i.e. level 0 is represented by r-vector which corresponding to the r bins of histogram; and level 1 is represented by $4r$-vector, etc. In our experiment, we follow [12] to set l to 3 and r to 20 in order to avoid over-fitting.

The PHOG descriptor is created by concatenating all the histograms: $P = (H_0, H_1, ..., H_{n-1})$, where H_0, H_1, …, H_{n-1} are the histogram-vector at each level respectively. Then, by concatenating all the PHOG descriptors we obtain the feature descriptor of the 3D model: $F = (P_1, P_2, ..., P_n)$, where $n = 18$ is the total number of the views.

IV. WEIGHT OPTIMIZATION

The dissimilarity measure between two 3D models A and B is defined as:

$$Dis(A,B) = d(F_A, F_B) = \sum_{i=1}^{N} w_i d(P_{A_i}, P_{B_i}) \tag{1}$$

where F_A and F_B denote the feature descriptors of two models respectively. $N = 18$ is the total number of the 2D views. w_i is the corresponding weight for the i-th view. P_{Ai} and P_{Bi} are the PHOG descriptors of the corresponding view. $d(.)$ is the distance of two descriptors which is measured by Kullback-Leibler (KL) divergence [13].

With the pre-classified information in database, we regard the models from the same categories as the relevant ones. Let Dr be the sum of the distance dr between the feature descriptors of model A and its j-th relevant models R (including A itself):

$$Dr = \sum_{j}^{M} dr_j \tag{2}$$

where M is the size of the relevant category . We aim to minimize Dr so that the relevant models can appear as closely as possible in the retrieval results. To solve this optimization problem, a Lagrange multiplier is employed. The Lagrange function is written as:

$$L(\boldsymbol{w},\lambda) = Dr - \lambda(\sum_{i=1}^{N}\frac{1}{w_i} - 1) \tag{3}$$

where $\sum_{i=1}^{N}\frac{1}{w_i} = 1$ is the constraint to make sure that w_i will not be equal to zero. Then, by solving the equation (4):

$$\frac{\partial L}{\partial w_i} = \sum_{j=1}^{M}\sum_{i=1}^{N} d(P_{A_i}, P_{R_i}) + \lambda w_i^{-2} = 0 \tag{4}$$

we obtain the optimization solution of :

$$\lambda^* = -\sum_{i=1}^{N} w_i D_i \tag{5}$$

$$w_i^* = \sum_{n=1}^{N}\sqrt{\frac{D_n}{D_i}} \tag{6}$$

where $D_i = \sum_{j=1}^{M}\sum_{i=1}^{N} d(P_{A_i}, P_{R_i})$. Let a vector $\boldsymbol{w} = (w_1^*, w_2^*, \ldots w_N^*)$ be an optimal weight vector corresponding to the N views of a 3D model. The vector $\boldsymbol{w}$ is associated with the feature descriptors of the model. By adapting this optimization approach to all of the U models in database, we obtain a set $\boldsymbol{W} = \{\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots \boldsymbol{w}_U\}$.

To reduce the storage space, we conduct a step of vector quantization on W. Initially, W is properly clustered by using the K-means algorithm. We use its centroid to represent each cluster. Since the feature descriptors of 3D models are associated with the weight vectors, likewise, they are also classified into K groups. Note that, the groups are different from the pre-classified categories of models in database. In classification, we classify the feature descriptors with respect to whether their associated weights are in the same clusters. In this way, the feature descriptors in the K-th group are associated with the same weight vector c_k. With the feature descriptors and their associated group labels, a nonlinear multi-class SVM is trained. We use the radial basis function (RBF) as the kernel function:

$$K(F_A, F_B) = \exp(-\gamma\|F_A - F_B\|^2) \tag{7}$$

where F_A and F_B denote the feature descriptors of two models respectively, and γ is the inverse width of RBF kernel.

Giving a query model, the group of its feature descriptors q is first predicted by the trained SVM classifier. Afterwards, we obtain its corresponding optimal weight vector, which is the centroid of the q-th cluster. Similarity measuring between the query and its object in database is then conducted by using their corresponding feature descriptors associated with the optimal weights.

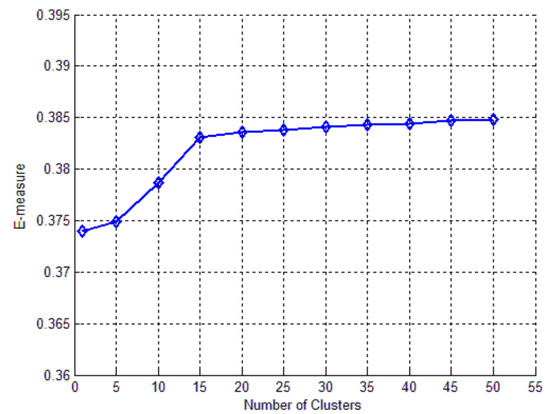


Figure 4. E-measure performance over different number of clusters

V. EXPERIMENT

We employ OpenCASCADE [14] as the visualization platform in the experiments. To evaluate the retrieval performance of the proposed method, 387 3D CAD models are selected from the Engineering Shape Benchmark (ESB) [2]. Those models are classified into 19 categories including gear-like parts, discs, cylindrical parts, etc. To ensure all of them are in the unified 3D coordinate system before feature extraction, the scale normalization is implemented using OpenCASCADE, and the rotation-normalization is carried out based on the PCA algorithm [6]. The 3D models are converted into the STEP formats required by OpenCASCADE. The images projected from the models are stored with the size of 256 by 256.

To evaluate the retrieval performance of our method, we employ the precision and recall (P-R) as the performance evaluation:

$$P = \frac{Ra}{Nr},\ R = \frac{Ra}{Ar} \tag{8}$$

where Ra denotes the number of retrieved relevant models contained in the top Nr matches. Ar denotes the total number of the relevant models.

We also use the E-measure [2] to measure the performance with a fixed retrieval size Nr. The E-measure is defined as:

$$E = \frac{(b^2+1)RP}{R + b^2 P} \tag{9}$$

where b denotes the relative importance of precision and recall, and is set to 0.5 in our experiments.

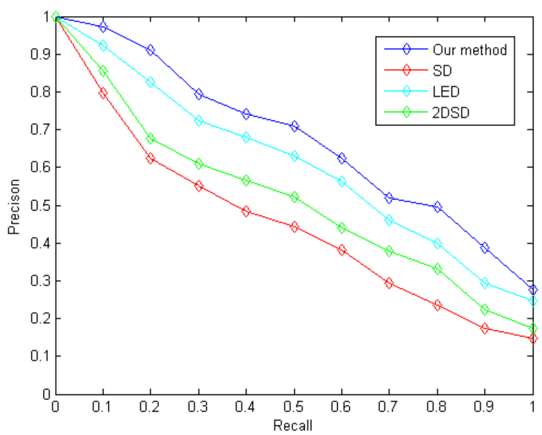


Figure 5. Precision and recall curve among different methods

TABLE I E-MEASURES ON THE FOUR METHODS

Method	E-measure		
	(Nr=10)	(Nr=20)	(Nr=32)
Our method	0.609	0.454	0.383
SD	0.501	0.347	0.281
2DSD	0.525	0.388	0.304
LFD	0.563	0.416	0.354

In our first experiment, we evaluate the influence of the number of clusters K on the retrieval performance. The value of K is changed within the range {1, 5, 10, … , 50}. Figure 4 illustrates the performance curve with different values of K. The performance is measured by E-measure with Nr = 32. As can be seen in the Figure, the performance has no obvious changes when the value of K is set from 15 to 50. Therefore, we set the value of K to be 15 for our following experiments. Note that, all of the 387 models are used for training the SVM. We set the free parameters of SVM by 8-fold cross-validation.* Our method achieves a cross validation accuracy of 92.74% when K is set to 15, i.e. the models are clustered into 15 clusters

In our second experiment, we compare the proposed method (where K is set to 15) with other three methods: 2D shape distribution (2DSD) [4], shape distribution (SD) [16], and Light field descriptor (LFD) [5].

From the result of the precision and recall curve (in Figure 5), it is obvious that our method achieves the best retrieval result. The result of the E-measure (in Table I) also shows the proposed method performs better than other competing methods.

VI. CONCLUSION

In this paper a new view-based method for 3D CAD model retrieval is proposed. It compares two models by summing up the similarity of their corresponding projected images associated with optimal weights. PHOG descriptor is introduced to represent the images and a novel scheme is proposed to adaptively assign each of the images an optimal weight. Our experimental results show that our method is superior to the state-of-the-art retrieval methods.

* LIBSVM [15] is used to train a SVM classifier; the values of the parameter C and γ are picked from the range {1, 2, … , 50}and {0.05, 0.10,… , 1}, respectively.

ACKNOWLEDGMENT

The authors would like to thank Jin Yu and Yanzhi Chen for their helpful comments and discussions. The authors also appreciate the support from China Scholarship Council.

REFERENCE

[1] N.Iyer, S.Jayanti, K.Lou, Y.Kalyanaraman and K.Ramani, "Three-dimensional shape searching: state-of-the-art review and future trends," *J. Computer-Aided Design*, vol.37, no.5, pp. 509-530, 2005.

[2] S.Jayanti, Y.Kalyanaraman, N.Iyer and K.Ramani, "Developing an engineering shape benchmark for CAD models," *J. Computer- Aided Design*, vol.28, no.9, pp. 939-953, 2006.

[3] P.Shilane and T.Funkhouser, "The Princeton Shape Benchmark," in *Proc. Shape Modeling Applications*, pp. 167-178, 2004.

[4] J.T.Pu. and K. Ramani, "On visual similarity based 2D drawing retrieval," *J. Computer-Aided Design*, vol.38, no.3, pp. 249-259, 2006.

[5] D.Y.Chen, X.P.Tian, Y.T.Shen and M.Ouhyoung, "On Visual Similarity Based 3D Model Retrieval," *J. Computer Graphics Forum*, pp. 223-232, 2003.

[6] D.Vranic, "3D Model Retrieval," Ph.D dissertation, University of Leopzig, 2004.

[7] R.Ohbuchi, K.Osada, T.Furuya and T.Banno, "Salient Local Visual Features for Shape-Based 3D Model Retrieval," in *Proc. SMI'08*, 2008.

[8] R.Ohbuchi. and F. Takahiko, "Scale-Weighted Dense Bag of Visual Features for 3 Model Retrieval from a Partial View 3D Model," in *Proc. ICCV 09 workshop on Search in 3D and Video(S3DV)*. 2009.

[9] T.F.Ansary, M.Daoudi, "A Bayesian 3-D Search Engine Using Adaptive Views Clustering," *J.* IEEE Transactions on Multimedia, vol.9, no.1, pp.78-88, 2007.

[10] T.F. Ansary, J.P.Vandeborre and M.Daoudi, "Bayesian approach for 3D models retrieval based on characteristic views," in *Proc. ICPR*, 2004.

[11] B.Hu, Y.Liu, S.Gao, J.Hu an R.Sun, "A powerful parial relevance feedback for 3D model retrieval," J. *Journal of multimedia*, vol.4. No.3. pp. 120-128, 2009.

[12] A.Bosch, A.Zisserman and X.Munoz, "Representing shape with a spatial pyramid kernel," in *Proc. CIVR,* pp.401-408, 2007.

[13] C.M.Bishop, Pattern Recognition and Machine Learning. Springer,2006

[14] OpenCASCADE Technology www.opencascade.org

[15] C.Hsu, C.Chang and C.Lin, "A practical guide to support vector classification,"2009[Online].Available:http://www.csie.ntu.edu.tw/~cjlin/libsvm/index.html

[16] C.Y.Ip and W.C.Regli, "Content-Based Classification of CAD Models with Supervised Learning," J. *Computer-Aided Design&Applications,* vol.2, no.5, pp. 609-617, 2005.