Computer Aided Drafting, Design and Manufacturing
Volume 23, Number 1, March 2013, Page 60

CADDM

Shape matching and retrieval based on multiple feature descriptors

WANG Weiming [1], *LIU Xiuping* [1], *LIU Ligang* [2]

1) School of Mathematical Sciences, Dalian University of Technology, 116023, China
2) School of Mathematical Sciences, University of Science and Technology of China, 230026, China

Abstract: A lot of 3D shape descriptors for 3D shape retrieval have been presented so far. This paper proposes a new mechanism, which employs several existing global and local 3D shape descriptors as input. With the sparse theory, some descriptors which play the most important role in measuring similarity between query model and the model in the dataset are selected automatically and an affinity matrix is constructed. Spectral clustering method can be implemented to this affinity matrix. Spectral embedding of this affinity matrix can be applied to retrieval, which integrating almost all the advantages of selected descriptors. In order to verify the performance of our approach, we perform experimental comparisons on Princeton Shape Benchmark database. Test results show that our method is a pose-oblivious, efficient and robustness method for either complete or incomplete models.

Key words: similarity; invariant; shape descriptor; subspace clustering

1. Introduction

With the development of the Internet, more and more 3D model databases can be acquired on the web [1-2] for free. Users can acquire the shape they want through the web or other ways conveniently [3]. At this time, how to accurately find the right model becomes an important issue. In order to solve this problem, content-based 3D shape retrieval arouses a strong concern in computer graphics, computer vision and pattern recognition community [4].

Many algorithms have been proposed for content-based 3D shape retrieval [5]. Most of them represent a shape with a feature vector. And then, a similarity metric is defined to measure the similarity distance between shapes, such as Euclidean distance, x^2 distance and Earth mover distance. With the feature vectors and similarity metric, a shape retrieval algorithm can be performed. That is to say, given a query shape, a shape retrieval algorithm will return the models with the descending order of similarity distance between the query model and the model in the dataset.

Most methods extract feature vectors according to the geometrical or topological properties of the models. With the extracted feature vectors, performance is evaluated by some common evaluation criterion, i.e. precision-recall (PR) curve [2]. These feature vectors are often invariant under different kinds of transformations, such as translation, rotation and non-rigid transformation, but none of them gives perfect results for all of the transformations. For some models in the different classes may have the same geometrical properties or topological properties, which leads some algorithms cannot distinguish these models. Inspired by the work of Hu et al. [6], we develop a mechanism by applying several descriptors existing at one time to construct our new descriptor. For a query, our method automatically selects several descriptors for different models in the dataset that play a key role in measuring the similarity of the query and this model. Descriptor obtained by our method integrates most of the advantages of the descriptors that are selected with sparse subspace clustering.

Our method is divided into four main steps:

Firstly, pose normalization. In order to make dissimilarity measure are invariant under scale, translation, rotation or non-rigid transformation, we first

Project Item: Supported by National Natural Science Foundation of China (61222206, 61173102, U0935004) and the One Hundred Talent Project of the Chinese Academy of Sciences.

Corresponding author: LIU Ligang, **E-mail:** lgliu@ustc.edu.cn

align all of the models in a canonical coordinate system using the principal component analysis (PCA) [7], and then multidimensional scaling (MDS) [8-9] is applied to eliminate the influence of the pose (see Fig.1).

Secondly, several existing global and local descriptors which are represented by histogram are calculated to form our feature space.

Thirdly, the sparse subspace clustering method [10] is applied to select some important descriptors automatically, which play crucial role in distinguishing between query model and the model in the dataset. At the same time, a similarity matrix will be generated.

At last, a spectral clustering method is used to obtain a new shape descriptor for each model which will be used to measure the similarity between the query model and the model in the dataset.

We compare our method with some other algorithms which only use a single feature vector. The results show that the retrieval performance of our method is the best. It can address the case of non-rigid or deformable shapes, which includes a wide range of shape transformations, such as, bending, articulated motion and case of incomplete models.

The main contributions of this work are:

Sparse subspace clustering is adopted to select several existing descriptors which play a key role in measuring the similarity between the query and the model in the dataset for different models and construct an affinity matrix;

Spectral clustering method is applied to obtain a feature vector for each model which integrates almost all the advantages of the selected descriptors.

The rest of the paper is organized as follows. Some related work will be introduced in section 2. In section 3, we will briefly introduce sub-space clustering and the detail of ours algorithm. Some experimental results and compared results will be shown in section 4. Finally, conclusion, limitations and future work will be recommended in section 5.

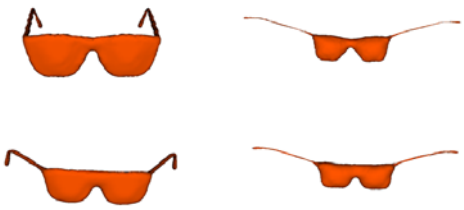

Fig. 1. Glasses with low dimensional embedding. The first column are two original glasses with different poses, the second column are the embedded results with multidimensional scaling (MDS)[8-9].

2. Related work

Great deals of algorithms have been proposed for 3D shape retrieval recently. For the detail of the recent review, we refer the reader to [5, 11] which give a detailed classification of existing methods. In the following, some related work will be introduced briefly.

2.1 Feature vector based global methods

Extended Gaussian Image (EGI) is proposed in [12], which is a spherical function and describes the distribution of surface normals. Complex Extended Gaussian Image (CEGI) proposed in [13] improves the EGI with complex-value spherical function, which not only gives the distribution of the surface normals, but also the associated normal distance of points on the surface. Shape function distribution [14] of the surface is used for the shape representation. The shape functions include the distance of a surface point to the shape central (D1), the distance between two random points (D2), the area of the triangle defined by three random point (D3) and the volume of the tetrahedron define by four random surface points (D4). In [15], three different methods are developed for representing 3D models. They are Shells, Sectors and the combination of Shells and Sectors. Poisson function is used in [16] to generate a histogram for each model, which capture shape structure feature very well. The methods described above are histogram based.

Chen et al. in [17] propose a descriptor named Light Field Descriptor (LFD), which represents a model as a collection of images rendered from uniformly sampled positions on a view sphere. The distance between two descriptors is defined as the minimum L1-difference, taken over all rotations and all pairings of vertices on two dodecahedra. Giorgi el al.[18] select a best view from a collection of views for the representation of 3D shape. They assume that the best view where the relevant shape features are maximally exposed can achieve a great performance. In [19], eigenvalues or eigenvectors of an appropriately defined affinity matrix are applied to represent a model. These descriptors work well for articulated 3D shapes. Funkhouse et al. [20] propose a web-based search engine which uses spherical harmonics to compute similarities and supports 2D and 3D sketches as queries.

Spherical harmonics, moments and 3D Fourier Transform are used in [21]. Hough Transform is applied by Zaharia et al. in [22] to define a descriptor named Canonical 3D Hough Transform Descriptor (C3DHTD). Novotni et al. [23] present a method to compute 3D Zernike descriptors from voxelized mod-

els as natural extensions of spherical harmonics based descriptors. Suzuki et al. [24] define an equivalence class for each voxelized model, and then a feature vector is constructed by the values in each equivalence class. The obtained descriptor is rotation invariant. As we can see that the descriptors mentioned above are consist of the coefficient of the transformation.

2.2 Feature vector based local methods

In [25], Gal et al. define their shape signature with the distribution of the local-diameter function (DF) which measures the diameter of the 3D shape in the neighborhood of each vertex and centricity function (CF) which measures the average geodesic distance from one vertex to all other vertices on the surface. Although two local shape descriptors are used, they are only combined into a 2D scalar-value array simply. Bronstein et al. [26-27] and Dey et al. [28] construct descriptors with heat kernel signatures [29], which depict the intrinsic property of the shape and are invariant under isometric deformations. These methods can handle scaled, partial or incomplete models.

The performance of some local shape descriptors are comparative studied in [30]. These local descriptors include the Distance to plane (DTP), Normal distribution (ND), Mean curvature (MC), Gaussian curvature (GC), Shape index (SI) and Curvature index (CI). They do experiments with each of these descriptors or the any combination of two of them. The results show that some combinations give the better results, while some of them give bad results. In [31], Funkhouser et al. estimate the retrieval performance of each local descriptor, and select most distinction shape descriptors for each query. This work is the most similar with our method. But it only uses the local descriptors and the selected descriptors are combined directly which cannot take advantage of all the advantages of descriptors, even some descriptors may play a negative impact. In this paper, some global and local descriptors existing will be used, and a sparse subspace clustering method will be applied to obtain a new feature vector for each model which integrates almost all the advantages of the descriptors selected.

2.3 Subspace clustering

Subspace clustering is used to represent high dimensional data set with several low dimensional data sets, and each of the low dimensional data set is a subspace of high dimensional data set (see Fig.2). For the detail of subspace clustering, we refer interested readers to the survey [10]. Our work is based on s sparse subspace clustering method, which selects a feature subspace in a given feature space.

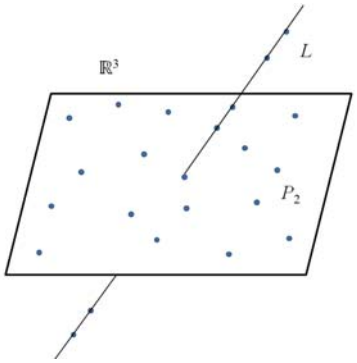

Fig. 2. A set of 3D points (blue) and two subspaces L and P_2 (black).

3. Sparse subspace clustering based shape descriptor

In this section, we will introduce the detail of our algorithm. Similar to Hu et al. [6], several selected local and global feature descriptors are used to construct an affinity matrix with sparse subspace clustering method, which selecting some descriptors that play an important role in identification of query model and the model in the dataset automatically. Then, our descriptor can be obtained by spectral clustering.

3.1 Preprocessing

3D models can be acquired from different ways, so they may have arbitrary scale, pose, and position in the 3D space. Because some of the descriptors are not invariant under scale, translation, or rotation, the normalization procedures may be necessary. The center of mass plays the crucial role on normalization procedure, which is defined as the barycenter of the model.

We scale the distance between the point on the model and the center of mass to a constant and translate the center of mass to the origin which normalize the scaling and translation.

The principal component analysis (PCA) method [7] is used to normalize the rotation in this paper. It aligns the principal axes to the x-, y-, and z-axes of a canonical coordinate system by an affine transformation based on the set of vertices of a 3D model. However, a variety of poses of the model which belong to the same class may cause the three principle axes very different. Therefore, we must normalize the pose of each model.

In order to normalize the pose of the model, multi-dimensional scaling (MDS) [8-9] is applied, which is known as an isometric embedding method and widely used in the general of isometry-invariant shape recognition and real-time texture mapping on complicated

3D objects. By implementing MDS for each model, the models within the same class will have the similar pose (see Fig.1).

Note that, all models can be aligned in a canonical coordinate system, which is befit for the computation of the feature vector in the following step.

Since our method uses several descriptors, we must select some descriptors existing which construct the feature space of our method. For sparse subspace clustering method requests the feature vectors must be the linear (affine) combination of the others, so the descriptors selected in this paper are histogram based. To our knowledge, histogram based descriptor has this characteristic. Although it restricts our choice of descriptors, the test results shown in the section 4 illustrate that histogram based descriptors can achieve appealing results.

3.2 Sparse subspace clustering

Before constructing our new descriptor, we will briefly introduce the method of sparse subspace clustering (SSC) [10].

SSC is based on the idea of representing a data point as a linear combination of any other points in the data set. In general, this causes an ill-posed problem with many possible solutions. To solve this problem, sparsity theory is introduced. Specifically, each point is represented as a sparse linear combination of all other points by minimizing the number of nonzero coefficients c_{jk} subject to $\mathbf{x}_j = \sum_{k \neq j} c_{jk}\mathbf{x}_k$. For N points, the optimization problem can be written as:

$$\min_{\{c_{jk}\}} \sum_{j=1}^{N} \sum_{k \neq j} |c_{jk}|, \quad \textbf{\textit{s.t.}} \quad \mathbf{x}_j = \sum_{k \neq j} c_{jk}\mathbf{x}_k \quad \text{and} \quad \sum_{k \neq j} c_{jk} = 1, \quad j = 1, \ldots, N \tag{1}$$

Eq.1 can be rewritten in matrix form as:

$$\min_{\mathbf{C}} \|\mathbf{C}\|_1 \quad \textbf{\textit{s.t.}} \quad \mathbf{X} = \mathbf{XC}, \quad \text{diag}(\mathbf{C}) = 0 \tag{2}$$

where, diag(**C**) represents the Diagonal of **C**.

In general, **X** = **XC** is considered as a soft constraint and thus the sparse coefficients can be obtained by solving the following optimization problem:

$$\min_{C} \|\mathbf{XC} - \mathbf{C}\|_F^2 + \lambda \|\mathbf{C}\|_1 \quad \textbf{\textit{s.t.}} \quad \mathbf{C} \geq 0, \ \text{diag}(\mathbf{C})=0 \tag{3}$$

where, $\|.\|_F$ denotes the Frobenius of matrix. The term $\|\mathbf{C}\|_1$ is used to guarantee the sparsity of the optimal solution **C***.

To solve Eq.3 efficiently, the optimization problem (3) is treated as a quadratic programming problem in [32]. They illustrate that their method can achieve the competitive accuracy as the results obtained by Eq.3. The quadratic programming formulation is:

$$\min_{C} f(\mathbf{C}) = \|\mathbf{XC} - \mathbf{C}\|_F^2 + \lambda \|\mathbf{C}^T\mathbf{C}\|_1 \tag{4}$$

$$\textbf{\textit{s.t.}} \quad \mathbf{C} \geq 0, \ \text{diag}(\mathbf{C}) = 0$$

Note that $\|\mathbf{C}^T\mathbf{C}\|_1$ is equivalent to $\mathbf{e}^T\mathbf{C}^T\mathbf{Ce}$, where e is an all-one vector. The optimization (4) can be solved simply and efficiently with Spectral Project Gradient method (SPG) [33]. After obtaining the sparse representation coefficients for each point, the affinity matrix is defined as:

$$\mathbf{A} = |\mathbf{C}| + |\mathbf{C}^T| \tag{5}$$

Then any spectral clustering method can be used to obtain the final clustering results.

3.3 Construction of descriptor

In section 3.1, some existing descriptors which are represented as histogram have been calculated. With N selected descriptors, we use Xi denotes a $m \times n$ feature matrix of the i-th descriptor, where m indicates the dimension of the i-th descriptor and n indicates the number of the models. That is to say, each column of the Xi indicates a feature vector of the corresponding model. We assume that if a model's feature vector is a linear combination of some other models', then these models are in the same class. Note that, since a model's feature vector can be combined with a small number of other models', the combination coefficients are sparse. So, our problem can be effectively solved by sparse subspace clustering theory. Eq.3 must be resolved for each descriptor. So they can be combined into a single optimization equation as follows:

$$\min_{\{\mathbf{C}_1,\ldots,\mathbf{C}_N\}} \sum_{i=1}^{N} f(\mathbf{C}_i) + g(\mathbf{C}_1,\ldots,\mathbf{C}_N) \tag{6}$$

$$\textbf{\textit{s.t.}} \quad \mathbf{C}_i \geq 0, \ \text{diag}(\mathbf{C}_i) = 0, \ i = 1, 2, \ldots, N$$

where $f(\mathbf{C}_i) = \|\mathbf{X}_i\mathbf{C}_i - \mathbf{C}_i\|_F^2 + \lambda \|\mathbf{C}_i^T\mathbf{C}_i\|_1$ is the object function for the i-th descriptor,

$$g(\mathbf{C}_1,\ldots,\mathbf{C}_N) = \alpha \|\mathbf{C}\|_{2,1} + \beta \|\mathbf{C}\|_1 \tag{7}$$

is the penalty function about $\mathbf{C}_1,\ldots,\mathbf{C}_N$. **C** is a $N \times n^2$ matrix, which is defined as:

$$\mathbf{C} = \begin{pmatrix} (\mathbf{C}_1)_{11} & (\mathbf{C}_1)_{12} & \cdots & (\mathbf{C}_1)_{n^2} \\ (\mathbf{C}_2)_{11} & (\mathbf{C}_2)_{12} & \cdots & (\mathbf{C}_2)_{n^2} \\ \vdots & \vdots & \ddots & \vdots \\ (\mathbf{C}_N)_{11} & (\mathbf{C}_N)_{12} & \cdots & (\mathbf{C}_N)_{n^2} \end{pmatrix} \tag{8}$$

and $\|.\|_{2,1}$ is the $l_{2,1}$ norm defined as $\|\mathbf{C}\|_{2,1} = \sum_{i=1}^{n^2} \|\mathbf{C}(:, i)\|_2$, $\mathbf{C}(:, i)$ denotes the i-th column vector of **C**, $\|.\|_2$ is the Euclidean norm, and the $\alpha \geq 0$ and $\beta \geq 0$ are two parameters which are used to balance the effect of the two penalty terms.

The penalty function $g(\mathbf{C}_1,\ldots,\mathbf{C}_N)$ is very important for our Algorithm. The first term penalizes the column sums of the **C**, so that the column sums of **C** as sparse as possible. That is to say, the combination coefficients are big if these models are in the same class; otherwise the coefficients are small or close to zero. Then from the affinity matrix defined in Eq.5, we can see that the similarity value will be big if the model and the query belong to the same class, or the similarity value will be small. The second term penalizes the overall sparsity of the **C**, which leads to the selection of the descriptors automatically. In other words, the feature vectors corresponding to the nonzero element of each column of the **C** constitute the subspace of the feature space, which play a key crucial role in identifying the query and the model in the dataset. For each model, the coefficients corresponding to the selected descriptors are big, and the other coefficients are small or close zero. Therefore, these two penalty terms take the full consideration of the similarity of the model and help us select the important descriptors to measure the similar of the query and the model in the dataset. In this way, our new descriptor can integrate the advantages of all descriptors.

In the previous stage, an affinity matrix **A** has been obtained by Eq.5. Next, any spectral clustering method can be applied to **A**, which not only returns the results of clustering but also returns the results embedded in the spectrum space. In our experiment, NCut method [34] is used to obtain the results embedded in the spectrum space, which forms our new descriptor. Then, any dissimilarity measure is used to measure the similarity of the query and the model in the dataset. We use Euclidean distance as our measure metric and use $\mathbf{V}_1$ and $\mathbf{V}_2$ denote feature vectors of two models in our paper. Then the similarity between these two models is defined as:

$$D(\mathbf{V}_1,\mathbf{V}_2)=\sqrt{\sum_{i=1}^{k}(\mathbf{V}_1(i)-\mathbf{V}_2(i))} \tag{9}$$

where, $\mathbf{V}_j(i)$ is the i-th element of the $\mathbf{V}_j$, j = 1, 2, k is the dimension of the feature vector.

4. Results

4.1 Experimental setup

Dataset. In our experiments, we use Princeton Shape Benchmark database [2]. In order to facilitate experiments, our dataset consisted of 320 models which are obtained from 18 model classes. Some models are incomplete. On average a model contains 2000 vertices and 3996 triangles. The pose of the models are very different. The query set consists of 34 shapes taken from different classes of the dataset.

Descriptors. In this paper, we use some existing global and local descriptors which are suitable for the subspace clustering method as our inputs. In the following, alldescriptors that are used in our experiments are listed as follows:

Curvature Index (CI) [5]: defined as:

$$\sqrt{\frac{\kappa_1^2+\kappa_2^2}{2}}$$

where κ_1 and κ_2 are the principal curvature at a point on the model.

Gaussian Curvature (GC) [5]: $\kappa_1\kappa_2$, where κ_1 and κ_2 are the principal curvature at a point on the model.

Centricity Function (CF or AGD) [25]: measures the average geodesic distance from one vertex to all other vertices on the model.

Extended Gaussian Image (EGI) [12]: a spherical function giving the distribution of surface normals.

D1 Shape Distribution (D1) [14]: a histogram of distances between the center of the mass of the model and the arbitrary vertex on the model.

Shape Histogram (SECSHEL) [15]: SECSHEL is the combination of the sector model and shell model, which measures the distribution of the vertices of the model. In this paper, we use 4 shell bins and 8 sector bins.

All descriptors depicted above are histogram-based descriptors which not only describe the global properties of the models but also the local properties. In our experiments, we use 100 bins for CI, Gauss, AGD and D1, 16 bins for EGI and 32 bins for SECSHEL.

Parameters. There are three parameters in our optimization function, which balance the effect of each penalty item. We set λ=1000, α=0.01, β=0.1 in all our experiments, which is not sensitive to different query models and always generates satisfactory results.

4.2 Evaluation

Precision-recall Curve[2]: For each query model in class S and any k top matches, "recall" (the horizontal axis) represents the ratio of models in class S returned within the top k matches, while "precision" (the vertical axis) denotes the ratio of the top k matches that are members of class S. Recall and Precision measures the effectiveness of a retrieval method. The final result is an average over all the models in the query set. A perfect retrieval result produces a horizontal line across the top of the plot (at precision = 1.0), indicating that the k models returned and the

query are belong to the same class. Otherwise, curves down.

Nearest Neighbor [2]: For each model in dataset, check whether the first match belongs to the same class as the query. The final result is an average over all the models in the query set. It is clearly that an ideal score is 100%, and higher scores indicate better results.

First/Second Tier [2]: The percentage of models in the query`s class that appear within the top $k$ matches, where $k$ depend on the size of the query`s class. For example, if the query`s class has n models, then $k = n$ -1 for the first tier and $k = 2\times(n - 1)$ for the second tier. The final result is the average over all the models in the query set. It is clearly that an ideal score is 100%, and higher scores indicate better results.

4.3 Our results

Some results of our algorithm will be shown in this section. For example, the retrieval results shown in Fig.3. The leftmost of the Fig.3(a) and Fig.3(c) are the query models (ant and monster) and the right five models are the first five matches which belong to same classes as the queries. These figures illustrate that our algorithm is robust to non-rigid transformation. Although the query of the Fig.3(c) is an incomplete model, the results are not affected. In other word, our method also can handle incomplete models. Fig.3(b) and Fig.3(d) are the similarity distance maps. The distances are scaled to between 0 and 1. They intuitively show the similarity distances between queries and the model in the dataset. The smaller the distance, the more similar of the two models. From these figures we can see that the similarity distances between the queries and the model in the queries' classes are smaller than the other models. These results fully show the robustness and effectiveness of our approach.

Fig.4 shows the top 10 matches for some queries. The images in the leftmost column show the query 3D models, while the columns on the right show the closest matches among the 3D models in the dataset. Obviously, for each query, all top 10 models returned and the query are belong to same class.

4.4 Comparison with other methods

In this section, we compare our method to some state-of-the-art methods for 3D shape retrieval. To evaluate the proposed method, each 3D model of the query set is used as a query object to retrieval the rest models of the dataset and a rank list is produced for each query. The final results are obtained by averaging over all the models in the query set.

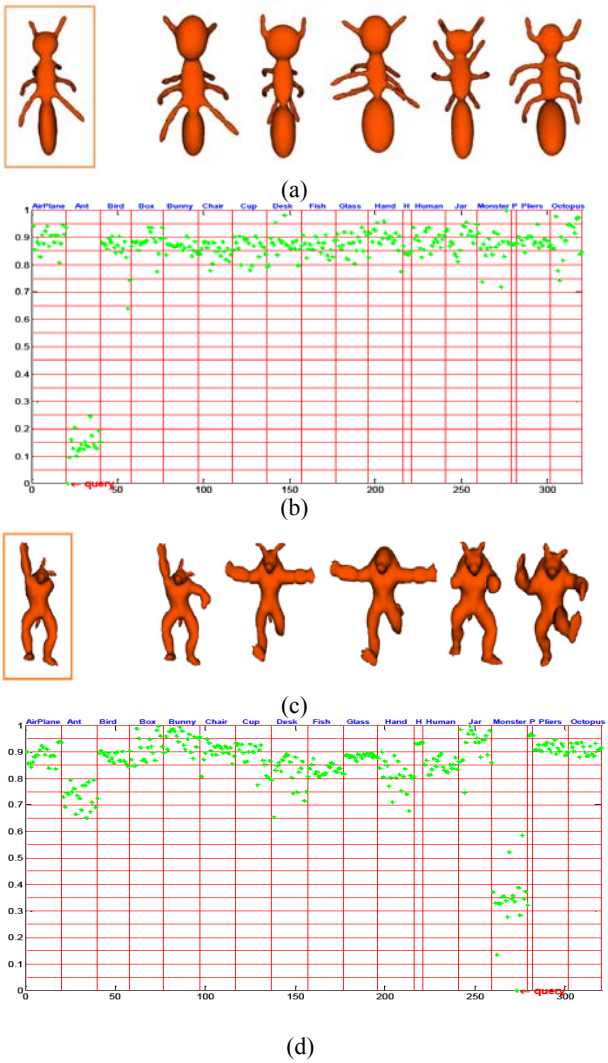

Fig. 3. Retrieval results of our approach. The leftmost of Figure (a) and (c) are the query models and the right five models are the first five matches. Figure (b) and (d) are the distance maps of the queries corresponding to Figure (a) and (c) respectively. Horizontal axis represents the model number and the vertical axis represents the similarity distance (green points) between the query model (red) and the model in the dataset. The top line of the figures mark with the class names (blue) of the models.

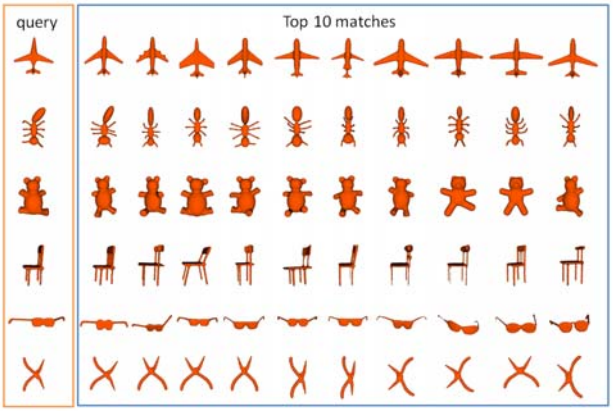

Fig. 4. Top 10 matches for each query of our method.

At first, we compare the performance of our method with the descriptors used in our algorithm which are depicted in subsection 4.1. Fig.5 is the Precision-Recall (PR) curves of the results which show

that our method obtains the best result (The top line of Fig.5). That is to say, any single descriptor cannot achieve our performance. The PR curve of our method almost close to level at precision equals to 1. Each descriptor has its own advantages and disadvantages. Our method integrates almost all the advantages of these descriptors.

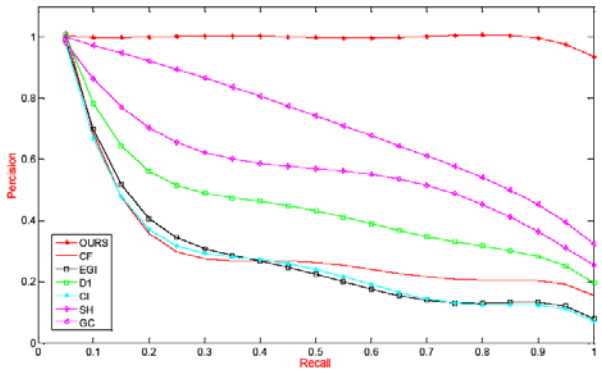

Fig. 5. Precision vs. recall curves for our method and the descriptors used in our approach.

Fig.6 shows the comparison of the retrieval performance of our approach with those of the others. They are Light Field Descriptor (LFD) [17], LFD used in spectral embedded space (new_LFD) [19], D2 shape distribution (D2) [14], Eigenvalues Descriptor (EVD) [19] and Poisson Histogram (PH) [16]. Each curve plots the graph of "precision and recall" averaged over all 320 classified models in the test dataset. From Fig.6 we can see that our curve is located above all of the curves which mean that our approach performs well than the others. Table 1 shows the nearest neighbor, and First/Second tier of these methods and our approach. The values measure the performance and accuracy of the methods.

We can see from the results of the comparison that our method is robust and efficient. It can not only deal with the incomplete models and nor-rigid models, but also obtain high retrieval accuracy.

4.5 Large dataset

Our method can handle large dataset. Given a query model and a large dataset, we divide the dataset into N subsets randomly so that each of the subsets contains only hundreds of models. The sparse subspace method is used on each subsets and query. Within each subset, the similarity distances between query model and the model in this subset are calculated and sorted in ascending order. The top k matches of each subset form a new dataset. Then the sparse subspace clustering method is used on this dataset again to obtain our final results.

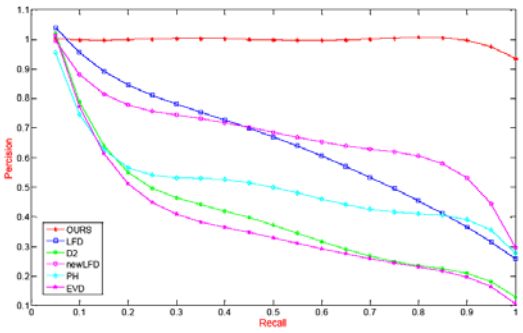

Fig. 6. Comparing the performance of our method with some other approaches.

Table 1. Nearest Neighbor and First/Second Tier of these methods and our method.

-	Nearest Neighbor	First Tier	Second Tier
Ours	100%	98.56%	49.75%
LFD	95.42%	51.67%	33.73%
New_LFD	81.82%	58.37%	35.89%
EVD	72.73%	28.71%	20.33%
PH	54.55%	45.93%	29.90%
D2	72.73%	30.62%	24.16%

5. Conclusions

In this paper, we proposed a new mechanism to use several existing descriptors. It automatically selects some descriptors which play a key role in measuring the similarity between the query and the model in the dataset by sparse subspace clustering theory. It produces a new descriptor which integrates almost all the advantages of these descriptors. Experimental results show that our algorithm is a pose-oblivious method which is very efficient and robust under complete and incomplete models.

However, our approach still has some shortcomings. Firstly, the descriptors selected must be histogram based. Because the histogram based descriptors have the characteristic of sparse representation and can be applied to the sparse subspace clustering theory. Secondly, our algorithm cannot handle the models of different scale, so all of the models must be scaled to the same scale in the preprocessing step.

In the future works, we plan to develop a new optimization function such that some of the descriptors that are not histogram based can be used. To deal with scale issues, we will utilize some descriptors that scale invariant, for example the descriptor proposed by Bronstein et al. in [26]. The models used in our paper are from Princeton database provided by Shilane et al.[2].

References

[1] AIM@SHAPE shape repository. http://shapes.aim-atshape.net/index.php.

[2] Shilane P, Min P, Kazhdan M, et al. The princeton shape benchmark [C] // Proceedings of Shape Modeling Applications, 2004. Proceedings IEEE, 2004: 167-178.

[3] Hitomi Y, Hideki A. Design Support Method Based on Analysis of Shape Impression [J]. Computer Aided Drafting, Design and Manufacturing, 2011, 21(2): 25-31.

[4] Meng X B, Tong B G, Pan Z J, et al. Modeling and Analysis of Transitional Tube with Constant Sectional Area along Derivative Central Route [J]. Computer Aided Drafting, Design and Manufacturing, 2011, 21(2).

[5] Tangelder J W H, Veltkamp R C. A survey of content based 3D shape retrieval methods [C] // Proceedings of Shape Modeling Applications, 2004. Proceedings IEEE, 2004: 145-156.

[6] Hu R Z, Fan L B, Liu L G. Co-Segmentation of 3D Shapes via Subspace Clustering. [C] // Computer Graphics Forum, 2012, 31(5): 1703-1713.

[7] Vranic D V, Saupe D, Richter J. Tools for 3D-object retrieval: Karhunen-Loeve transform and spherical harmonics [C] // Proceedings of Multimedia Signal Processing, 2001 IEEE Fourth Workshop on IEEE, 2001: 293-298.

[8] Bronstein M M, Bronstein A M, Kimmel R, et al. Multigrid multidimensional scaling [J]. Numerical linear algebra with applications, 2006, 13(2-3): 149-171.

[9] Rosman G, Bronstein A M, Bronstein M M, et al. Topologically constrained isometric embedding [C] // Proceedings of Human Motion. Springer Netherlands, 2008: 243-262.

[10] Vidal R. A tutorial on spectral clustering [C] // IEEE Signal Processing Magazine, 2011, 28(2): 52-68.

[11] Bustos B, Keim D A, Saupe D, et al. Feature-based similarity search in 3D object databases [C] // Proceedings of ACM Computing Surveys (CSUR), 2005, 37(4): 345-387.

[12] Horn B K P. Extended gaussian images [C] // Proceedings of the IEEE, 1984, 72(2): 1671-1686.

[13] Sing B K, Katsushi I. Determining 3-D Object Pose Using The Complex Extended Gaussian Image [C] // Proceedings of Computer Vision and Pattern Recognition, 1991. Proceedings CVPR'91., IEEE Computer Society Conference on. IEEE 1991: 580-585.

[14] Osada R, Funkhouser T, Chazelle B, et al. Shape distributions [C] // ACM Trans. Graph. (TOG), 2002, 21(4): 807-832.

[15] Mihael A, Gabi K, Hans P K, et al. 3D shape histograms for similarity search and classification in spatial databases [C] // Proceedings of Advances in Spatial Databases. Springer Berlin Heidelberg, 1999: 207-226.

[16] Pan X, You Q, Liu Z, et al. 3D shape retrieval by Poisson histogram [J]. Pattern Recogn. Lett, 2011, 32(6): 787-794.

[17] Che D Y, Tian X P, Shen Y T, et al. On visual similarity based 3d model retrieval [C] // Computer Graphics Forum, 2003, 22(3): 223-232.

[18] Giorgi D, Mortara M, Spagnuolo M. 3D shape retrieval based on best view selection [C] // Proceedings of the ACM workshop on 3D object retrieval. ACM. New York, NY, USA: 2010: 9-14.

[19] Jain V, Zhang H. A spectral approach to shape-based retrieval of articulated 3D models [J]. Comput. Aided Des. 2007, 39(5): 398-407.

[20] Funkhouser T, Min P, Kazhdan M, et al. A search engine for 3D models [J]. ACM Trans. Graph. 2003, 22(1): 83-105.

[21] Vrani c D V, Saupe D. 3D Shape Descriptor Based on 3D Fourier Transform [C] // Proceedings of the EURASIP conference on digital signal processing for multimedia communications and services (ECMCS 2001) , Budapest, Hungary, 2001.

[22] Zaharia T, Prêteux F. Shape-based retrieval of 3D mesh models [C] // Proceedings of Multimedia and Expo, 2002. ICME'02. 2002 IEEE international Conference on IEEE, 2002, 1: 437-440.

[23] Cicirello V, Regli W C. Machining Feature-Based Comparisons of Mechanical Parts [C] // Proceedings of the International Conference on Shape Modeling & Applications. SMI'01.Washington, DC, USA: IEEE Computer Society: 2001: 176-185.

[24] Suzuki M T, Kato T, Otsu N. A Similarity Retrieval of 3D Polygonal Models by Using Rotation Invariant Shape Descriptors [C] // Proceedings of Systems, Man, and Cybermetics, 2000 IEEE international Conference on IEEE, 2000, 4: 2946-2952.

[25] Gal R, Shamir A, Cohen-Or D. Pose-Oblivious Shape Signature [C] // IEEE Transactions on Visualization and Computer Graphics, 2007, 13(2): 1077-2626.

[26] Bronstein A M, Bronstein M M, Guibas L J, et al. Shape google: Geometric words and expressions for invariant shape retrieval [J]. ACM Trans. Graph., 2011, 30(1): 1-20.

[27] Ovsjanikov M, Bronstein A M, Guibas L J, et al. Shape Google: a computer vision approach to invariant shape retrieval [C] // Proceedings of Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on. IEEE, 2009: 320-327.

[28] Dey T K, Li K, Luo C, et al. Persistent Heat Signature for Pose-oblivious Matching of Incomplete Models [C] // Computer Graphics Forum, 2010, 29(2): 1545-1554.

[29] Sun J, Ovsjanikov M, Guibas L J. A concise and provably informative multi-scale signature based on heat diffusion [C] // Computer Graphics Forum, 2009, 28(5): 1383-1392.

[30] Sarah T, Afzal G. An evaluation of local shape descriptors for 3D shape retrieval [C] // Proceedings of SPIE. 2012, 8290: 82900N.

[31] Shilane P, Funkhouser T. Selecting Distinctive 3D Shape Descriptors for Similarity Retrieval [C] // Proceedings of Shape Modeling and Applications, 2006. SMI 2006. IEEE International Conference on. IEEE, 2006: 18-27.

[32] Wang S, Yuan X, Yao T, et al. Efficient subspace segmentation via quadratic programming [C] // Proceedings of Innovative Applications of Artificial Intelligence Conference. 2011, 1: 519-524.

[33] Birgin E G, Martłnez J M, Raydan M. Nonmonotone spectral projected gradient methods on convex sets [J]. SIAM Journal on Optimization, 2000, 10(4): 1196-1211.

[34] Shi J, Malik J. Normalized cuts and image segmentation [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2000, 22(8): 888-905.