

CM-BOF: visual similarity-based 3D shape retrieval using Clock Matching and Bag-of-Features

Zhouhui Lian · Afzal Godil · Xianfang Sun ·
Jianguo Xiao

Received: 19 July 2012 / Revised: 11 March 2013 / Accepted: 18 March 2013
© Springer-Verlag Berlin Heidelberg 2013

Abstract Content-based 3D object retrieval has become an active topic in many research communities. In this paper, we propose a novel visual similarity-based 3D shape retrieval method (CM-BOF) using Clock Matching and Bag-of-Features. Specifically, pose normalization is first applied to each object to generate its canonical pose, and then the normalized object is represented by a set of depth-buffer images captured on the vertices of a given geodesic sphere. Afterwards, each image is described as a word histogram obtained by the vector quantization of the image's salient local features. Finally, an efficient multi-view shape matching scheme (i.e., Clock Matching) is employed to measure the dissimilarity between two models. When applying the CM-BOF method in non-rigid 3D shape retrieval, multidimensional scaling (MDS) should be utilized before pose normalization to calculate the canonical form for each object. This paper also investigates several critical issues for the CM-BOF method, including the influence of the number of views, codebook, training data, and distance function. Experimental results on five commonly used benchmarks demonstrate that: (1) In contrast to the traditional Bag-of-Features, the

time-consuming clustering is not necessary for the codebook construction of the CM-BOF approach; (2) Our methods are superior or comparable to the state of the art in applications of both rigid and non-rigid 3D shape retrieval.

Keywords 3D shape retrieval · Non-rigid · Bag-of-Features · Local feature

1 Introduction

How to efficiently and effectively retrieve 3D models based on their shapes has become an active topic in several research communities including computer vision [30,46], pattern recognition [52,40], computer graphics [38,10], and multimedia [56,12]. With the development of various kinds of 3D shape retrieval benchmarks (e.g., PSB [49], ESB [20], McGill [50], NSB [14], etc.) and the successful organization of the 3D SHape REtrieval Contest (SHREC) [2], more and more researchers have been attracted to work in this area and a large number of algorithms have been proposed.

Feature extraction plays an important role in 3D shape retrieval methods. Ideally, a good shape descriptor has the following desirable properties: (1) High discrimination; (2) Efficient shape matching; (3) Compact representation; (4) Efficient feature extraction; (5) Invariance to similarity transformations (for non-rigid 3D shape retrieval, descriptors should also be isometry-invariant [46]); (6) Invariance to shape representation; (7) Invariance to shape degeneracies and noises. Generally speaking, based on the shape descriptors used, existing 3D shape retrieval methods can be loosely classified into four categories [11]: statistics-based, graph-based, transform-based, and view-based. Recent investigations [9,11,30,37] show that view-based methods with pose normalization preprocessing obtain significantly better

Z. Lian (✉) · J. Xiao
Institute of Computer Science and Technology, Peking University,
Beijing, People's Republic of China
e-mail: lianzhouhui@yahoo.com.cn

J. Xiao
e-mail: xiaojianguo@pku.edu.cn

A. Godil
National Institute of Standards and Technology,
Gaithersburg, USA
e-mail: godil@nist.gov

X. Sun
Cardiff University, Wales, UK
e-mail: Xianfang.Sun@cs.cardiff.ac.uk

performance in retrieving rigid models than other approaches and more importantly they satisfy almost all characteristics mentioned above. Among all kinds of view-based approaches, the visual similarity-based method which follows the idea that “if two 3D models are similar, they also look similar from all viewing angles” is considered to be the most powerful one. Therefore, despite having several intrinsic drawbacks (e.g., discarding invisible information of an object), visual similarity-based approaches are still without doubt the most popular and practical methods in the field of 3D shape retrieval.

Probably because of the high computational complexity of shape matching for local features, existing visual similarity-based methods all utilize global shape descriptors to represent 2D views, which hinders the further improvement of retrieval performance. As a matter of fact, local features have been widely used in many computer vision applications [35] and usually methods that employ local features result in better performance than other methods using only global features. Consequently, it is reasonable to infer that similar computer vision techniques can be applied into content-based 3D object retrieval, especially for the view-based 3D shape retrieval methods.

In this paper, we propose a new visual similarity-based 3D shape retrieval approach (CM-BOF) using Clock Matching and Bag-of-Features. More specifically, the method describes each view as a word histogram built by vector quantization of the view’s salient local descriptors and employs an efficient multi-view shape matching scheme to compute the dissimilarity between 3D objects. An overview of the CM-BOF method is presented as follows: First, a 3D model is properly aligned to the canonical coordinate frame so that the normalized pose could be well suited to draw standard three-view images and then depth-buffer views are captured on the surrounding vertices of a unit geodesic sphere. Afterwards, for each view we extract salient local features (e.g., SIFT [33]) which are subsequently quantized into a word histogram using the Bag-of-Features approach. Finally, according to the properties of the geodesic sphere previously used, an efficient shape matching is carried out to measure the dissimilarity between two objects by computing the minimum distance of all (24) possible matching pairs. To the best of our knowledge, our work [28] is the first to employ local features in the visual similarity-based method for 3D shape retrieval, and this article is the extended version of the conference paper. However, our previous method [28] can only deal with rigid 3D models (see Fig. 1 for some examples). In fact, existing visual similarity-based methods are all essentially unsuitable to distinguish and recognize non-rigid objects. This is because, as shown in Fig. 2, even if two articulated 3D models are generated from the same object, they still may look quite different from many viewing angles. This paper extends our previous work [28,29] to make the method also



Fig. 1 Examples of rigid 3D models that are classified into the same category



Fig. 2 Examples of non-rigid 3D models that are classified into the same category

well suited for the retrieval of non-rigid 3D shapes. Specifically, when applying the CM-BOF approach to retrieve non-rigid 3D objects, multidimensional scaling (MDS) should be utilized before pose normalization to calculate the canonical form for each object. By doing this, excellent performance can be obtained by the proposed method in applications of both rigid and non-rigid 3D shape retrieval.

The major contributions of this paper are twofold.

1. A novel visual similarity-based 3D shape retrieval framework is proposed, where the Bag-of-Features method is utilized to describe each view as a word histogram and the objects are compared by an efficient multi-view shape matching scheme. Moreover, using MDS embedding, the proposed method can also obtain excellent performance in the application of non-rigid 3D shape retrieval.
2. Exhaustive experiments are carried out carefully to investigate the influence of the number of views, codebook, training data, and distance function. Perhaps surprisingly, our results show that, in contrary to the traditional Bag-of-Features, the time-consuming clustering is not necessary for the codebook construction of our method.

The rest of the paper is organized as follows. Section 2 discusses previous work. Section 3 presents a detailed description of our method. Experimental results are then shown and analyzed in Sect. 4. Finally, we provide the conclusion of this paper in Sect. 5.

2 Related work

Based on the shape descriptors used, existing 3D shape retrieval methods can also be classified into the following two categories: global feature-based methods and local feature-based methods. For more information about the development in 3D shape retrieval, we refer the reader to a recent survey [56].

2.1 Global feature-based 3D shape retrieval

Most of the existing 3D shape retrieval methods belong to this category. So far, a large number of 3D global shape descriptors have been proposed such as D1 [3], D2 [38], spherical harmonic descriptor (SHD) [23], 3D wavelet descriptor [24], skeleton descriptor [53], Reeb graph descriptor [4], light field descriptor (LFD) [10], DESIRE [59], and so on. Since our 3D shape descriptor is designed to be able to measure the visual similarity between two objects, we pay more attention to the visual similarity-based methods, which have been considered as the most discriminative approaches in the literature [2, 49].

Among these visual similarity-based methods, LFD [10] method is perhaps the most famous one. In the LFD method, each 3D model is represented by 100 silhouettes (10 views per group) rendered from uniformly distributed viewpoints on the half side of a unit sphere and the silhouette is encoded by a feature vector consisting of 35 Zernike moments and 10 Fourier coefficients. They measured the dissimilarity between two objects by the minimum distance of 6,000 view group matching pairs, considering all possible situations. LFD method avoids pose normalization via an exhaustive searching which inevitably aggravates computational cost. To address this problem, Lian et al. [30] developed a multi-view shape matching scheme for properly normalized generic models. The experiments showed that, with the same image descriptors, retrieval performance including discrimination, spatial requirement and searching speed could be considerably improved compared to the original LFD method. Also, Daras et al. [12] achieved accurate rotation estimation using the combination of plane reflection symmetry and rectilinearity to normalize each 3D object, and then represented the model as a set of 2D binary view images. In each image, 2D Polar-Fourier coefficients, Zernike moments and Krawtchouk moments are extracted to generate the view's shape descriptor. As a matter of fact, pose normalization has been widely applied in many visual similarity-based 3D shape retrieval methods [8, 9, 42, 44, 47]. The major difference between them is the feature vectors they used to describe views. For instance, 2D Fourier coefficients [42], the elevation descriptor [47], and the depth-line descriptor [9] have been employed to represent depth-buffer views. Similarly, silhouette views have been described using the 2D shape distribution descriptor [44] and 1D Fourier coefficients [8]. The methods discussed above all utilized the images captured from viewpoints located on the sphere. Recently, Papadakis et al. [41] proposed a 3D shape descriptor using a set of panoramic views, which were obtained by projecting a 3D model to the lateral surface of a cylinder. The panoramic views are described by their 2D Discrete Fourier coefficients as well as 2D Discrete Wavelet coefficients. Papadakis [39] further improved the retrieval performance of the method

[41] using a local relevance feedback technique that basically shifts the 3D shape features closer to their centroid in feature space.

2.2 Local feature-based 3D shape retrieval

2D local features have been proven to be very successful in many applications (e.g., image retrieval [21], object classification [15], video data mining [51], etc.) and a vast number of 3D local features (e.g., 3D spin image [22], harmonic shape context [16], 2.5D SIFT [32], etc.) have also been developed. However, considerably less work has been reported to apply local features in 3D shape retrieval. This is mainly due to the high computational cost of shape matching for huge amounts of local descriptors extracted from 3D objects. Local feature-based 3D shape retrieval is an interesting and promising research topic, since it has intrinsic capability of solving non-rigid 3D shape retrieval [26] and partial 3D shape retrieval problems [17]. Funkhouser and Shilane [17] selected distinctive multi-scale local features, which are calculated via spherical harmonic transformation, and applied priority-driven search to efficiently achieve partial matching. Gal et al. [19] proposed a curvature-based local feature that describes the geometry of local regions on the surface and then constructed a salient geometric descriptor by clustering together a set of local descriptors which are interesting enough according to a given saliency function. Geometric hashing was utilized to accelerate the partial matching of salient local features. Tal and Zuckerberge [55] decomposed each object into meaningful components, and then, based on the decomposition, they represented the 3D model as an attributed graph that is invariant to non-rigid transformations. Wang et al. [61] proposed intrinsic spin images (ISIs) generalizing the traditional spin images [22] from 3D space to N-dimensional intrinsic shape space, in which ISIs shape descriptors are computed from MDS embedding representations of original 3D shapes. More recently, Lian et al. [26] made a comparison of methods for non-rigid 3D shape retrieval and found that a large percentage (more than 60 %) of these state-of-the-art approaches utilize local features to represent non-rigid 3D objects.

Bag-of-Features, which is a popular technique to speed up the matching of image local features, has recently been introduced into local feature-based 3D shape retrieval. Liu et al. [31] presented a compact 3D shape descriptor named "shape topics" and evaluated its application to 3D partial shape retrieval in their paper [31], where a 3D object is represented as a word histogram constructed by quantizing the local features of the object. Spin images, calculated on points randomly sampled on the surface, are chosen as the local descriptors. Li et al. [25] introduced a weak spatial constraint to the method proposed in [31] by partitioning a 3D model into different regions based on the clustering of local features' spatial positions, but the improvement was limited.

Toldo et al. [57] applied a more sophisticated mesh segmentation method to decompose a 3D object into several subparts. Each segmented region is represented by one descriptor and then a word histogram is generated by assigning all subpart descriptors of the object into visual words. More recently, Bronstein et al. [6] employed multiscale diffusion heat kernels as “geometric words” and used the “Bag-of-Features” approach to construct compact and informative shape descriptors for 3D models. They also demonstrated that considering pairs of “geometric words” allows creating spatially sensitive bag of features with improved discriminative power.

The work that is most relevant to our paper is [37], in which Ohbuchi et al. reported a view-based method using salient local features (SIFT [33]). They represented a whole 3D object by a word histogram derived from the vector quantization of salient local descriptors extracted on the depth-buffer views captured uniformly around the object. Their experiments demonstrated that the method resulted in excellent retrieval performance for both articulated and rigid objects. To some extent, the CM-BOF method proposed in this paper is quite different from the BF-SIFT algorithm presented in [37]. Basically, our approach is a visual similarity-based method, following the idea that “if two 3D models are similar, they also look similar from all viewing angles”; while BF-SIFT [37] is a “global Bag-of-Features based method”, which represents a 3D model as a single word histogram via the vector quantization of its local features. Moreover, several new techniques have been developed to make our method be well suited for practical applications of both rigid and non-rigid 3D shape retrieval, and results also demonstrate that our method could markedly outperform BF-SIFT [37] in terms of retrieval accuracy.

3 Method description

In this section, we first present an overview of our method and then elaborate on the details of each step in the corresponding subsection.

3.1 Overview

Since the method is mainly based on the *Bag-of-Features* approach and a multi-view shape matching scheme (named *Clock Matching* for the sake of convenience and intuition), we call it “CM-BOF” algorithm in this paper. The CM-BOF algorithm, depicted in Fig. 3, is implemented subsequently in four steps:

1. *Shape preprocessing* Normalize 3D objects with respect to the canonical coordinate frame to ensure that their mass centers coincide with the origin, they are bounded by the

unit sphere, and they are well aligned to three coordinate axes. For the application of non-rigid 3D shape retrieval, before pose normalization, the canonical form of each object is calculated using MDS.

2. *Local feature extraction* Capture depth-buffer views on the vertices of a given unit geodesic sphere whose mass center is also located in the origin and then extract salient local features from these range images.
3. *Word histogram construction* For each view, quantize its local features into a word histogram using a pre-specified codebook. Normally, the codebook is obtained off-line by clustering the training data that are randomly sampled from the feature set of all models in the target database. However, the codebook of our method can also be directly built using randomly sampled N_w local feature vectors. This has been verified by the experiments described later.
4. *Clock matching* Carry out an efficient multi-view shape matching scheme to measure the dissimilarity between two 3D models by calculating the minimum distance of their 24 matching pairs.

3.2 Shape preprocessing

As shown in Fig. 3, for the application of non-rigid 3D shape retrieval, the 3D canonical form of each object should be computed before the procedure of pose normalization. While, for rigid 3D shape retrieval, we only need to normalize the pose for the original model. Details of the *pose normalization* and *canonical form computation* are presented in Sects. 3.2.1 and 3.2.2, respectively.

3.2.1 Pose normalization

Since the key idea of our method is to measure the visual similarity between 3D objects with arbitrary poses, it is preferable if all models can be well normalized to the canonical coordinate frame in the same manner. Therefore, we normalize the objects by a recently proposed approach [30] which combines the PCA (principal component analysis) based and the rectilinearity based pose alignment algorithms to obtain better normalization results. As we know, PCA (here we use the continuous PCA [60]) is the most prominent tool for accomplishing rotation invariance by solving three principal axes of a 3D object. While the basic idea of the rectilinearity-based method (only suitable for polygonal meshes) is to specify a standard pose through the calculation of the model’s rectilinearity value. Three steps of the composite method are described as follows.

1. *Translation and scaling* For a given 3D mesh, translate the center of its mass to the origin and then scale the

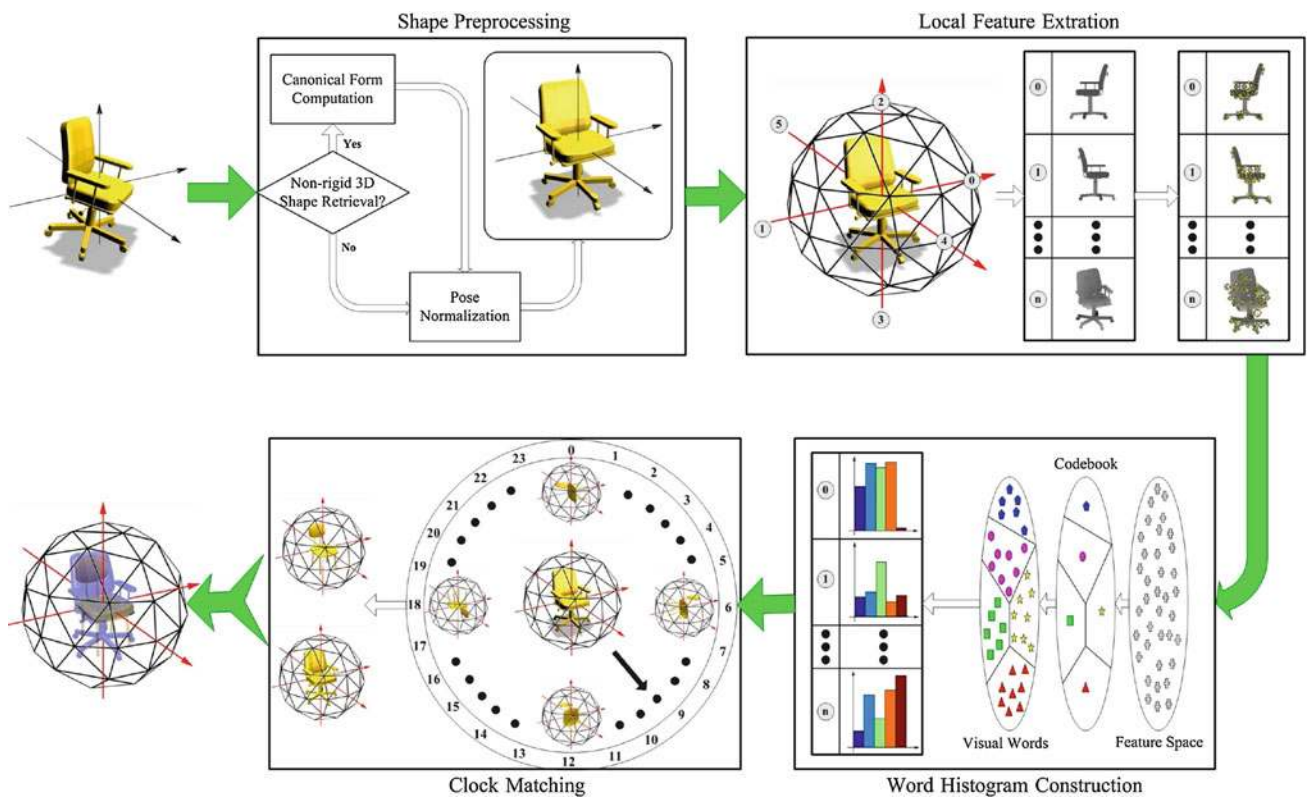


Fig. 3 An illustration of our method. First, the pose of a given 3D model is properly normalized. For non-rigid 3D shape retrieval applications, MDS embedding is applied to generate the canonical form of the object before pose normalization. Second, depth-buffer views are captured from the vertices on a given geodesic sphere and then, for each view, we calculate SIFT descriptors [33] on salient points. Third,

a word histogram is obtained by vector quantizing the view’s local features against the codebook, so that the object can be described as a set of histograms. Finally, an efficient shape matching (i.e., Clock Matching) is carried out to obtain the best match from all 24 matching pairs between two objects

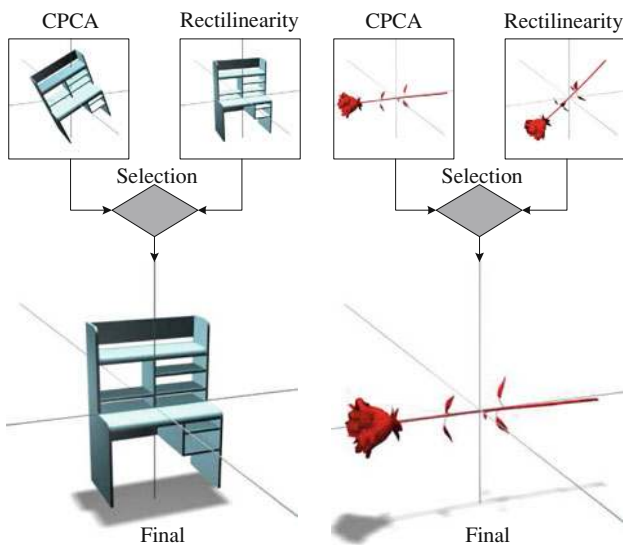


Fig. 4 Two alignment examples of the pose normalization method we use. The final result is chosen, using a selection criterion, from the normalization results of two methods

1. *Rotation by one method* Apply the PCA-based method, respectively, to rotate the original model to the canonical coordinate frame and then store these two normalized meshes in memory.
2. *Rotation by two methods* Apply the PCA-based and the rectilinearity-based method, respectively, to rotate the original model to the canonical coordinate frame and then store these two normalized meshes in memory.
3. *Selection* Calculate the number of valid pixels of three silhouettes, projected on the planes YOZ , ZOX , XOY , for the two normalized meshes generated in the previous step. And then select the model, which yields the smaller value, as the final normalization result.

Two normalization examples are displayed in Fig. 4. Note that, the method only performs incomplete pose normalization for rotation transformation. More specifically, only the positions of three axes are fixed for the model normalized by the composite method, that is, the direction of each axis is still undecided and the x -axis, y -axis, z -axis of the canonical coordinate system can be located in all three axes. That also means 24 different orientations are still plausible for

the aligned models, or rather, 24 matching operations should be carried out when comparing two normalized objects. For more details of the pose normalization algorithm, we refer the reader to our previous paper [30] where convincing experimental results have been obtained to illustrate the advantage of this approach, in the context of pose normalization and 3D shape retrieval, against other methods.

3.2.2 Canonical form computation

Non-rigid objects are commonly seen in our surroundings. Take Fig. 5a for an example, a human being might appear in several distinct postures that could inevitably be identified as different kinds of objects using most existing methods. In order to properly and efficiently measure the dissimilarity between two non-rigid objects, it is preferable that the models can be represented as some shape descriptors which are invariant or approximately invariant under isometric transformations (e.g., bending and articulation). Unless otherwise specified, isometric transformation mentioned in this paper means the transformation which preserve the geodesic distance between every pair of corresponding points on the surface. Based on the fact that the geodesic distance between every two points on a surface remains unchanged under isometric transformations, a bending invariant representation (i.e., *3D canonical form*) can be obtained by applying MDS to map the geometric structure of the surface to a new 3D Euclidean space, in which geodesic distances are approximated by Euclidean ones. This idea is originally proposed in [13], where three different MDS techniques are also compared. Examples of 3D canonical forms obtained using Classical MDS and Least Squares MDS are shown in Fig. 5b and c, respectively. It should be point out that typically MDS cannot be used in the application of rigid 3D shape retrieval. This is mainly due to the fact that models derived from the same object after applying different isometric transformations might be classified into different categories in the case of rigid shape retrieval, while the MDS embedding generates isometry-invariant representations for those models. For example, human models in the PSB database are classified into the following three categories: human beings in normal poses, walking, and with their arms out, respectively. If we apply MDS on these models, very similar canonical forms might be obtained and thus most probably they will be classified into the same category. That could obviously reduce the performance of rigid 3D shape retrieval methods.

The basic idea of MDS techniques is to map the dissimilarity measure between every two features in a given feature space into the distance between corresponding pair of points in a low-dimensional Euclidean space. More specifically, MDS map each feature Y_i , $i = 1, 2, \dots, N$ to its corresponding point X_i , $i = 1, 2, \dots, N$ in a m -dimensional

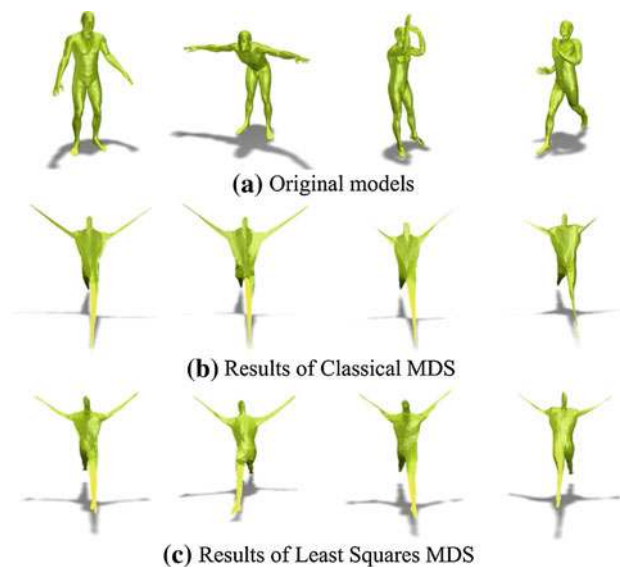


Fig. 5 Non-rigid models (a) and their 3D canonical forms obtained by applying Classical MDS (b) and Least Squares MDS (c), respectively

Euclidean space \mathbf{R}^m by minimizing, for example, the following stress function:

$$E_S(X) = \frac{\sum_{i=1}^N \sum_{j=i+1}^N w_{ij} (d_F(Y_i, Y_j) - d_E(X_i, X_j))^2}{\sum_{i=1}^N \sum_{j=i+1}^N (d_F(Y_i, Y_j))^2}, \quad (1)$$

where $d_F(Y_i, Y_j)$ denotes the dissimilarity between the feature Y_i and Y_j , $d_E(X_i, X_j)$ denotes the Euclidean distance between two points (i.e., X_i and X_j) in \mathbf{R}^m , w_{ij} is the weighting coefficient, and $X = [X_1, X_2, \dots, X_N]^T$. Specifically, in this paper, Y_i and Y_j stand for a pair of points on the original 3D mesh, while X_i and X_j denote the corresponding points on its canonical form. For our purpose of generating 3D canonical forms, the dimension of the Euclidean space is chosen as $m = 3$ and the geodesic distance $d_G(Y_i, Y_j)$ is selected as the dissimilarity measure between the pair of points on the original surface.

A standard optimization algorithm to solve the minimization problem of cost functions like $E_S(X)$ (Eq. (1)) is the Least Squares technique. However, it is not easy to calculate the closed expression for the first derivative of this nonlinear function. A simple but effective solution is to use the numerical computing technique with iterative majorization. The idea is applied in the SMACOF (scaling by maximizing a convex function) [5] algorithm to minimize the stress function $E_S(X)$. As demonstrated by experimental results in [13], the Least Squares MDS method using SMACOF obtained better minimization (see [13]) for the stress function (1) compared to other MDS techniques. Thereby, we choose to apply the Least Squares MDS embedding with the

SMACOF algorithm in our method. Here, we briefly review the SMACOF algorithm (more details can be found in [5]).

Minimizing the stress function $E_S(X)$ is equivalent to minimizing the following function:

$$E'_S(X) = \sum_{i=1}^N \sum_{j=i+1}^N w_{ij}(d_F(Y_i, Y_j) - d_E(X_i, X_j))^2 \quad (2)$$

or

$$E'_S(X) = \varphi_F^2 + \varphi_E^2(X) - 2\phi(X), \quad (3)$$

where

$$\varphi_F^2 = \sum_{i=1}^N \sum_{j=i+1}^N w_{ij}d_F^2(Y_i, Y_j), \quad (4)$$

$$\varphi_E^2(X) = \sum_{i=1}^N \sum_{j=i+1}^N w_{ij}d_E^2(X_i, X_j), \quad (5)$$

$$\phi(X) = \sum_{i=1}^N \sum_{j=i+1}^N w_{ij}d_F(Y_i, Y_j)d_E(X_i, X_j). \quad (6)$$

Applying the Cauchy–Schwartz inequality and some basic algebraic operations, we have

$$E'_S(X) \leq \varphi_F^2 + \text{trace}(X^T \Gamma X) - 2\text{trace}(X^T B(\tilde{X})\tilde{X}) = \sigma(X, \tilde{X}) \quad (7)$$

where \tilde{X} is the approximation of X , and the elements of the matrix $B(\tilde{X})$ are defined by

$$b_{ij} = \begin{cases} -\frac{w_{ij}d_F(Y_i, Y_j)}{d_E(\tilde{X}_i, \tilde{X}_j)}, & i \neq j \text{ and } d_E(\tilde{X}_i, \tilde{X}_j) \neq 0 \\ 0, & i \neq j \text{ and } d_E(\tilde{X}_i, \tilde{X}_j) = 0 \end{cases} \quad (8)$$

$$b_{ii} = \sum_{j=1, j \neq i}^N b_{ij}, \quad (9)$$

and the $N \times N$ matrix Γ is given by

$$\Gamma = \sum_{i=1}^N \sum_{j=i+1}^N w_{ij}E_{ij} \quad (10)$$

$$E_{ij} = (e_i - e_j)(e_i - e_j)^T, \quad (11)$$

where e_i is the vector that occupies the i th column of a $N \times N$ identity matrix.

Let the derivative of $\sigma(X, \tilde{X})$ be 0, that is

$$\frac{\partial \sigma(X, \tilde{X})}{\partial X} = 2\Gamma X - 2B(\tilde{X})\tilde{X} = 0, \quad (12)$$

we get the minimum of $\sigma(X, \tilde{X})$. Finally, the result of the minimization problem can be computed by

$$X^{(k)} = \Gamma^+ B(\tilde{X})\tilde{X}, \quad (13)$$

where Γ^+ is the Moore–Penrose inverse of Γ . By setting all weights w_{ij} to 1, Eq. (13) can be rewritten as

$$X^{(k)} = \frac{1}{N} B(X^{(k-1)})X^{(k-1)}. \quad (14)$$

In practice, given a threshold ε , calculating Eq. (14) iteratively until $E'_S(X^{(k)}) - E'_S(X^{(k-1)}) < \varepsilon$, we obtain the final solution $X^{(k)}$ for the nonlinear minimization problem of the stress function $E_S(X)$. Here, we use the matlab source code that is publicly available on the web site of the book [7] to implement the SMACOF algorithm.

As the calculation of geodesic distances and the SMACOF algorithm are both computationally expensive, 3D meshes should be simplified before the MDS embedding procedure. In this paper, the number of vertices on the simplified mesh is experimentally chosen as 2000, and it takes about 120 s on average to construct the canonical form for a 3D model under our experimental settings.

3.3 Local feature extraction

After pose normalization, 3D meshes (or their canonical forms) have been well aligned to the canonical coordinate frame. Then, we capture their depth-buffer views on the vertices of a given unit geodesic sphere whose mass center is also located in the origin. The geodesic spheres used here are obtained by subdividing the unit regular octahedron in the way shown in Fig. 6. These kinds of polygonal meshes are suitable for our multi-view based shape retrieval mechanism, mainly because of the following three reasons. First, the vertices are distributed evenly in all directions. Second, these geodesic spheres enable different levels of resolution in an intuitive manner. The coarsest (level-0) one is obtained using a unit regular octahedron with 6 vertices and 8 faces. Higher levels can be generated by recursive subdivisions. Third, since all these spheres are derived from an octahedron, given the positions of six vertices for the original octahedron, other vertices can be specified automatically. Moreover, all vertices are symmetrically distributed with respect to the coordinate frame axes. That means, when comparing two models, only 24 matching pairs need to be considered for the feature vector in an arbitrary level.

After view rendering, a 3D object can be approximately represented by a set of depth-buffer images from which we extract salient SIFT descriptors, as presented in [33]. The SIFT descriptor is calculated, using the *VLFeat* matlab source code developed by Vedaldi and Fulkerson [58], in the

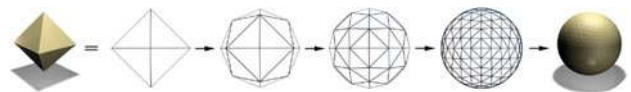


Fig. 6 Geodesic spheres generated from a regular octahedron

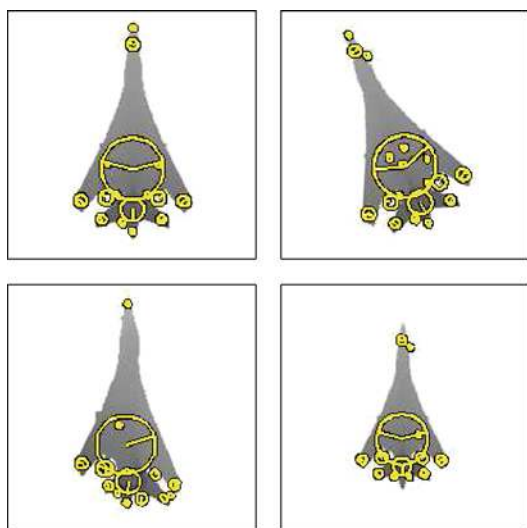


Fig. 7 A demonstration on the robustness of the salient local descriptor against small viewpoint changes

following two steps. First, obtain the scale, orientation, and position information of the salient points detected by the different-of-Gaussian (DoG) approach. Second, compute SIFT descriptors for the local interesting regions which are determined by the scale and position of the salient points. Here, the SIFT descriptor, which is actually a 3D histogram of gradient location and orientation, is computed using its default parameters, where the location is divided into a 4×4 grid and the gradient angle is quantized into eight orientations. This results in a feature vector with 128 elements. The feature is designed to be robust, to some extent, against similarity transformation, affine distortion, noise and illumination changes of images. Figure 7 shows some examples of SIFT descriptors extracted from the range images which are scaled, rotated, and affine transformed. It can be seen that the SIFT descriptor is stable to various changes of 3D viewpoints, which is a desirable property for our visual similarity-based 3D shape retrieval method to compensate its dependence on the stability of pose normalization.

3.4 Word histogram construction

Directly comparing 3D models (or their canonical forms) by their local visual features is time-consuming, especially for the 3D shape retrieval methods using a large number of views. To address this problem, we quantize the SIFT descriptors extracted from a depth-buffer image into one word histogram so that the view can be represented in a highly compact and discriminative way.

Before vector quantization, a codebook (also named as *vocabulary*) with N_w visual words should be created. Usually, the codebook is generated via off-line clustering. More specifically, huge numbers of feature vectors are first

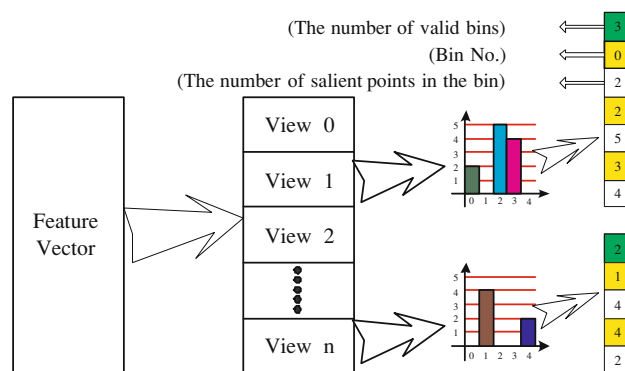


Fig. 8 The data structure of our 3D shape descriptor that is composed of several word histograms

randomly sampled from the feature set of the target database to form a training set. Then, the training set is clustered into N_w clusters using the K-means algorithm. At last, centers of the clusters are selected as the feature vectors of visual words in the codebook. Since the spatial requirement and calculating time of the K-means clustering are significant, many other algorithms [18] (e.g., kd-tree, ERC-tree, and Locality sensitive hashing) have been applied to alleviate the computational cost. While, as we can see from the experiments described in Sect. 4.2, clustering is not necessary for the codebook construction of our method. In other words, randomly sampled local feature vectors can be directly used to create the vocabulary and these two codebook construction approaches result in almost the same discriminative power for 3D shape retrieval.

By searching for the nearest neighbor in the codebook, a local descriptor is assigned to a visual word. Then, each view can be represented using a word histogram whose i th bin records the number of i th visual words in the depth-buffer image. To obtain satisfactory discrimination capability, usually the histogram should have thousands of bins. Let the number of views be 66 and the number of visual words in the codebook be 1,500, without optimization, the 3D shape descriptor would be of dimension 99,000. In fact, with the observation that, for our method, the average number of salient points in a view (with size 256×256) is only about 30, we can represent the histogram in a better way that not only makes the shape descriptor highly compact but also significantly improves the efficiency of dissimilarity calculation.

Figure 8 shows an example of the data structure for our 3D shape descriptor, where only the information (i.e., bin No. and bin value) of some bins, whose values are not equal to zero, appears in the feature vector. Experimental results show that, considering the method with 66 views and a 1,500-dimensional codebook, on average the new data structure requires about 30 times less spatial storage and performs approximately 21 times faster for feature comparison.

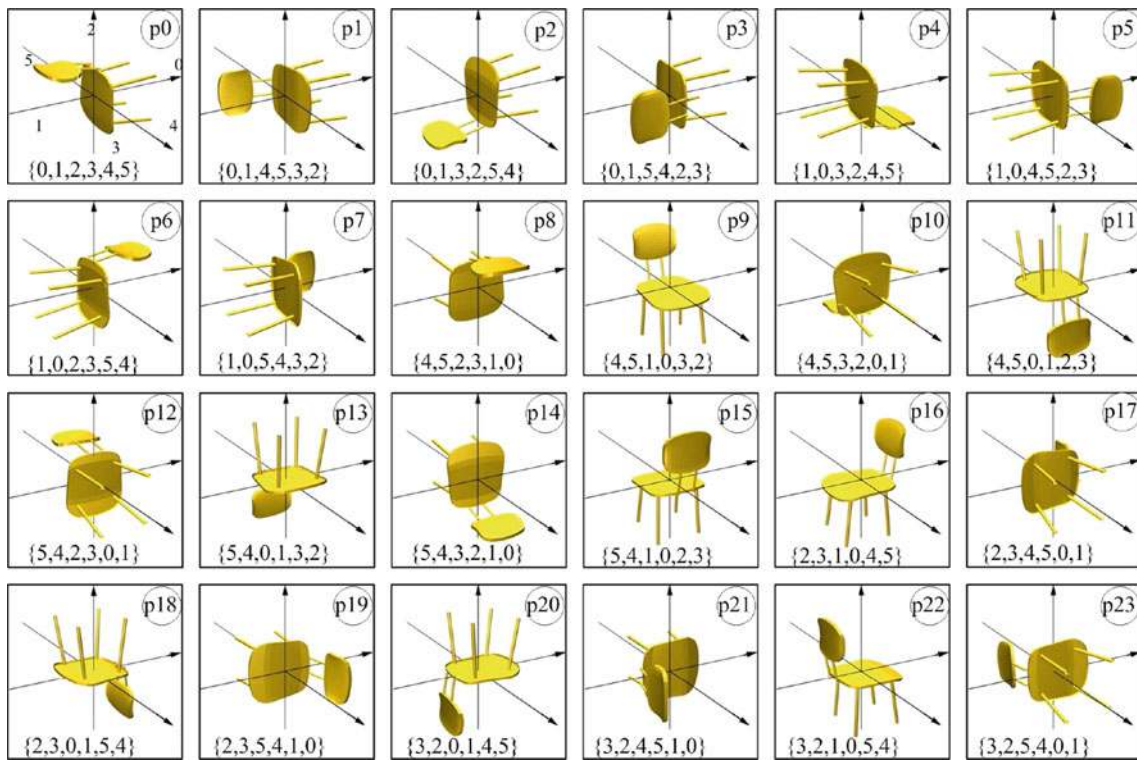


Fig. 9 All (24) possible poses of a chair after it has been incompletely aligned to the canonical coordinate frame. The corresponding permutations are listed underneath. The permutation $p_i, i = 0, 1, \dots, 23$ denotes the positions of three major axes of the object in the context of pose i

3.5 Clock Matching

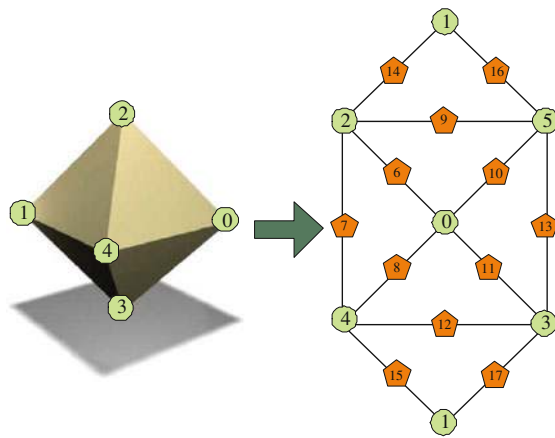
The last step of our 3D shape retrieval method is the dissimilarity calculation (also called *shape matching*) between two shape descriptors. The multi-view shape matching (i.e., *Clock Matching*) scheme we use was originally proposed in our previous paper [30] (almost simultaneously, Daras and Axenopoulos also reported the similar idea in [11]), here we provide more details about this approach and apply it into our shape matching task with new shape descriptors and new distance measures.

The basic idea of Clock Matching is that, after we get the major axes of an object, instead of completely solving the problem of fixing the exact positions and directions of these three axes to the canonical coordinate frame, all possible poses are taken into account during the shape matching stage. The principle of the method is simple and reasonable, moreover, our previous experiments [30] have already illustrated considerable improvements against other approaches. As we mentioned above, 24 different poses still exist for a normalized model. Figure 9 shows all possible poses of a chair after pose alignment processing in the canonical coordinate frame. For the sake of convenience, $x+, x-, y+, y-, z+,$ and $z-$ axis are denoted as 0, 1, 2, 3, 4, and 5, respectively. When comparing two models, one of them will be

placed in the original orientation denoted as a permutation $p_0 = \{p_0(k) | k = 0, 1, 2, 3, 4, 5\}$ while the other one may appear in 24 different poses denoted as permutations $p_i = \{p_i(k) | k = 0, 1, 2, 3, 4, 5\}, 0 \leq i \leq 23$. From these 24 permutations (see the underneath of each small image in Fig. 9), all possible matching pairs $((p_0, p_i), 0 \leq i \leq 23)$ between two models can be obtained. For instance, we can capture six silhouettes or depth buffers from the vertices of a unit regular octahedron and then extract 2D shape descriptors for these images to construct a view-based 3D feature vector. The vertices in the corresponding axes are also denoted as 0, 1, 2, 3, 4, and 5, respectively. Then, we compare all 24 matching pairs for two models and the minimum distance is chosen as their dissimilarity.

Generally speaking, *Clock Matching* performs in the following two steps:

1. *Initialization* Recursively subdividing the original unit octahedron n_d times, we get a geodesic sphere with the required resolution and the coordinates of its vertices should be recorded consecutively according to the time they emerge. During the process of subdivision, two tables (named *edge table* and *vertex table*, respectively) which indicate the relationship between old and new vertices are also obtained. An example of the *edge table* and



Edge No.	Vert No1.	Vert No2.
0	0	2
1	4	2
2	0	4
3	5	2
4	0	5
5	0	3
6	4	3
7	5	3
8	1	2
9	1	4
10	1	5
11	1	3

Edge Table

Vert No.	0	1	2	3	4	5
0	-1	-1	6	11	8	10
1	-1	-1	14	17	15	16
2	6	14	-1	-1	7	9
3	11	17	-1	-1	12	13
4	8	15	7	12	-1	-1
5	10	16	9	13	-1	-1

Vertex Table

Fig. 10 The edge table and the vertex table generated when subdividing the original octahedron into the geodesic sphere with 18 vertices. The indexes of the vertices of the original edges are stored in the edge table. New vertices' indexes can be obtained using pairs of old vertices

to search in the vertex table. A more intuitive illustration of the relations between the original vertices (green circles) and the new vertices (red pentagons) is given in the middle of this figure (color figure online)

the *vertex table*, utilized to store the information during the stage of subdividing the octahedron, is demonstrated in Fig. 10. Note that we only need to process this step once.

2. *Comparison* As mentioned above, when comparing two models represented by level-0 descriptors, we calculate the minimum distance among 24 matching pairs $((p_0, p_i), 0 \leq i \leq 23)$ which can be derived using the permutations shown in Fig. 9. If higher-level shape descriptors are applied, we should use the *edge table*, *vertex table*, and $p_i, 0 \leq i \leq 23$ to build new permutations $p'_i = \{p'_i(k) | 0 \leq k < N_v\}, 0 \leq i \leq 23$ describing all possible matching pairs $(p'_0, p'_i), 0 \leq i \leq 23$ for two models represented by N_v views. Finally, the dissimilarity between the query model q and the source model s is defined as

$$Dis_{q,s} = \min_{0 \leq i \leq 23} \sum_{k=0}^{N_v-1} D(FV_q(p'_0(k)), FV_s(p'_i(k))), \tag{15}$$

where $FV_m = \{FV_m(k) | 0 \leq k < N_v\}$ denotes the shape descriptor of 3D object m , $FV_m(k)$ is the signature of view k , and $D(\cdot, \cdot)$ is the distance function. In Sect. 4.4, four distance functions, denoted as D_{MaxHis} , D_{MinHis} , D_{AvgHis} , and D_{L1} , are defined and compared. By default, we utilize D_{MaxHis} to measure the dissimilarity between two views.

4 Experiments

In this section, we first present and discuss experimental results to study the influence of the number of views,

codebook, training data, and distance function on retrieval performance for our CM-BOF algorithm. Then, 3D shape retrieval results are analyzed for the visual similarity-based methods (CM-BOF and GSMD [30]) that use local features and global features, respectively. Finally, we compare the retrieval accuracy of our methods with the state of the art in applications of rigid and non-rigid 3D shape retrieval, respectively.

Experiments are carried out on the following five publicly available 3D shape benchmarks:

- *PSB* The test set of the Princeton Shape Benchmark [49] contains 907 generic models which are classified into 92 categories. The maximum number of objects in a class is 50, while the minimum number is 4.
- *NSB* The NIST (National Institute of Standards and Technology) Shape Benchmark [14] is composed of 800 generic models which are classified into 40 categories. Each class contains 20 objects.
- *GWSB* The Generic Warehouse Shape Benchmark [43] includes 3,168 generic models which are classified into 43 categories. The number of objects in each category varies between 11 and 177.
- *McGill* The McGill Shape Benchmark [50] consists of 255 articulated objects which are classified into 10 categories. The maximum number of objects in a class is 31, while the minimum number is 20.
- *SHREC'11 Non-rigid 3D Shape Benchmark* The Benchmark [27] contains 600 articulated 3D watertight meshes which are classified into 30 categories. Each class has 20 models.

More specifically, the first five experiments are conducted on the PSB and NSB databases, while experiments described

in Sect. 4.6 are carried out on all these five databases. We implement the feature extraction in Matlab R2007, and write the shape matching code in C++ using Microsoft Visual Studio 2005. All programs are run under windows XP on a personal computer with a 2.66 GHz Intel Core2 Quad CPU, 4.0 GB DDR2 memory, and a 128 MB NVIDIA Quadro Fx550 graphics card.

Note that: Unless otherwise specified, the default parameters of our CM-BOF method are selected as follows: the resolution of depth-buffer images is 256×256 , the number of views $N_v = 66$, the size (i.e., the number of visual words) of the codebook $N_w = 1,500$, the size (i.e., the number of local feature vectors) of the training set $N_t \approx 120,000$, the codebook is generated by clustering the training set, which is derived from the target database, using the *Integer k-means* method whose source code is available on the website [58].

4.1 Influence of the number of views

In the first experiment, we investigate the influence of the number of views on retrieval performance.

Figure 11 shows the precision-recall curves calculated for our CM-BOF methods using geodesic spheres with 6, 18, 66, and 258 vertices. It can be observed that retrieval performance can be improved by increasing the number of views, especially when the number of views jumps from 6 to 66. But the improvements slow down as the number of views keeps growing, while the computational cost still increases sharply. This is because an upper limit exists for the retrieval performance of view-based methods, but more views involved always means that more time needs to be spent on calculating local descriptors and more memories are required to store the feature vectors. Consequently, in order to make the balance between quality and cost, the num-

ber of views is experimentally chosen as 66 in the following sections.

4.2 Influence of the codebook

In the second experiment, we study the influence of the codebook size and creation methods by comparing retrieval performance among the shape descriptors corresponding to codebooks with different sizes and different construction methods.

Codebook size Probably, we can say that the most important parameter of the CM-BOF algorithm is the number of visual words (denoted as N_w) in the codebook. This is because the codebook size not only determines the spatial requirement but also significantly affects the retrieval performance of the method. Figure 12 shows results which report the discounted cumulative gain (DCG, well-known as the most stable retrieval measure [48]) values for CM-BOF methods with steadily increased codebook size. We observe that, as the codebook size enlarges, DCG values go up sharply at the beginning and become stable approximately when $N_w > 1,000$. Similar conclusions are obtained from Fig. 12 for the BF-SIFT method presented in [37], where only one word histogram is used to describe a 3D object. According to the experimental results presented here, we set the number of visual words in the codebook as 1,500 in this paper.

Construction methods Next, two codebook building methods are compared. The first one selects the centers of feature clusters to form the codebook, after the clustering of the train data set which is composed of a large number of local descriptors randomly sampled from the database to be retrieved. The second method directly uses the randomly sampled feature vectors as the visual words in the codebook. Typically, the Bag-of-Features method is implemented with clustering.

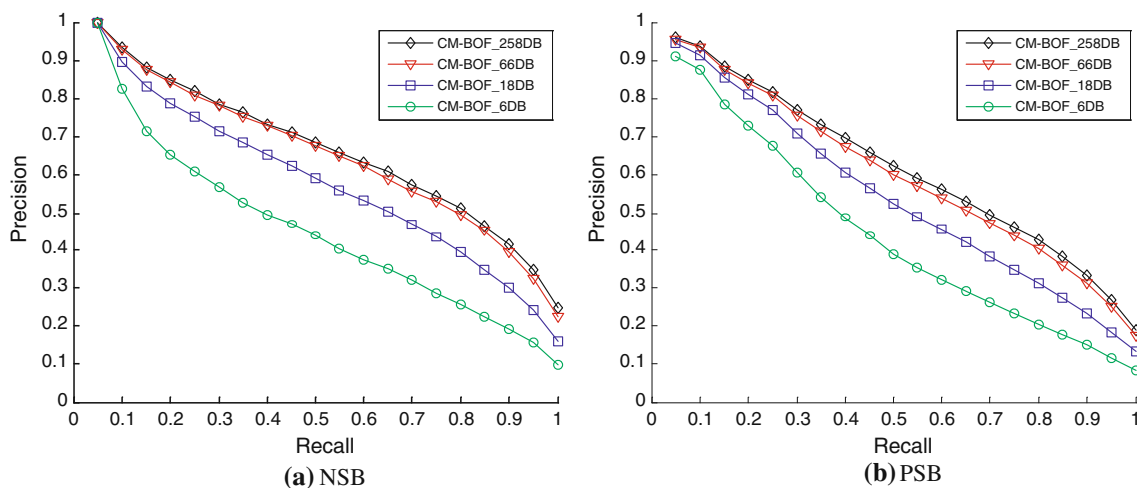


Fig. 11 Influence of the number of views. **a, b** Show the Precision-recall plots for the methods, with different numbers of views, run on the NSB and PSB databases, respectively

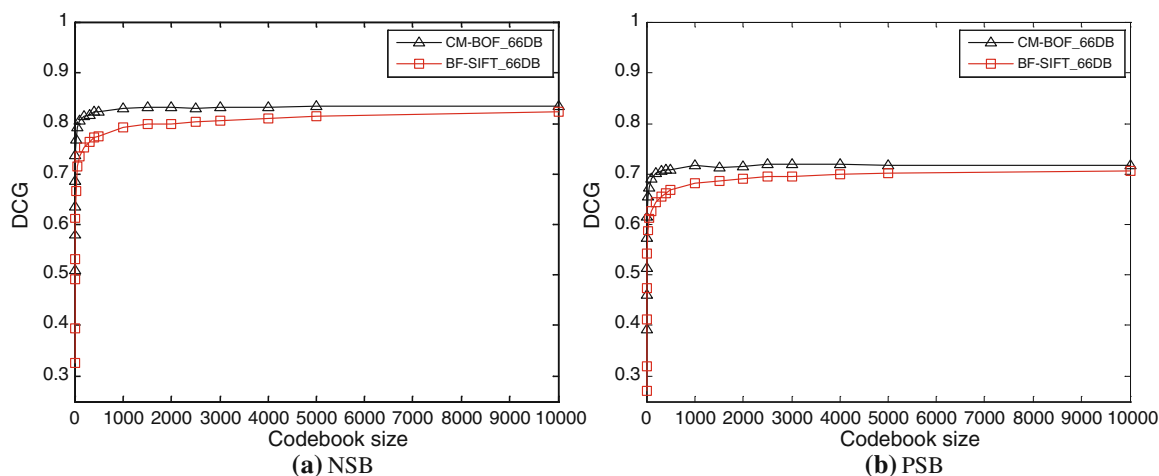


Fig. 12 Influence of the codebook size. **a, b** Show the DCG versus codebook size curves for two methods (i.e., CM-BOF and BF-SIFT [37]) run on the NSB and PSB databases, respectively

Table 1 Influence of the codebook construction method

	K-means (%)	Random (%)
NSB	83.1 ± 0.1	83.1 ± 0.2
PSB	71.7 ± 0.2	71.7 ± 0.2

The table gives the means and standard deviations of the DCG values over 10 runs of our CM-BOF algorithms on two benchmarks, for codebooks generated using clustering (K-means), and for randomly sampled codebooks (Random)

The previous work [36] has also demonstrated that the first codebook construction method results in better performance in image classification against the second method. However, from Table 1, which presents the means and standard deviations of the DCG values over 10 runs of our CM-BOF algorithms with and without clustering, a conclusion can be made that the random sampling approach works as well as the clustering approach for our CM-BOF 3D shape retrieval algorithm when the codebook size has been properly chosen. We infer that this is mainly due to the carefully designed shape matching scheme and the fewer invalid information existing in the views captured from the 3D objects compared to the ordinary images used in other 2D applications.

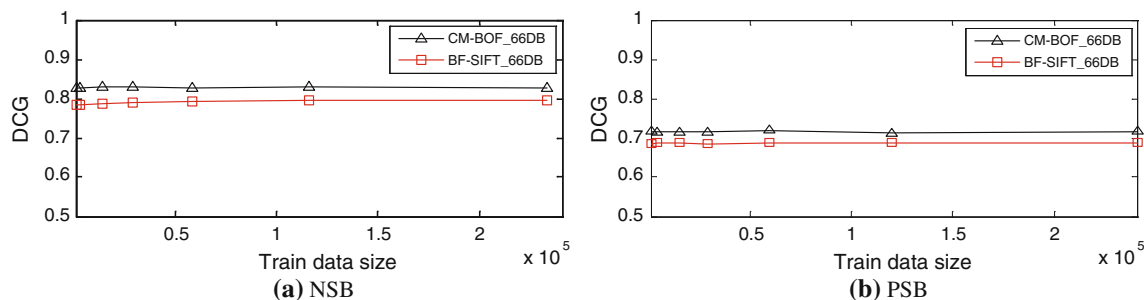


Fig. 13 Influence of the training data size. **a, b** Show the DCG versus training data size curves for two methods (i.e., CM-BOF, BF-SIFT [37]) run on the NSB and PSB databases, respectively

4.3 Influence of the training data

In the third experiment, we analyze the influence of the training data size and training source on retrieval performance.

Training data size Figure 13 shows the curves depicting the relation between DCG values and the number of feature vectors in the training set. It can be seen that the size of the training data has very little impact on the retrieval performance. No matter how large or how small the training set is, the corresponding retrieval performance remains stable, even when the size of the training data is just a little bit larger than the codebook (e.g., $N_w = 1,500$ and $N_t = 1,578$). The experimental results provide an additional support to the aforementioned conclusion that clustering is not necessary for our method.

Training data source It is worthy of investigating whether it is necessary to create the training set by sampling the feature vectors in the database to be searched. In other words, we want to study the influence of the source database from which the local descriptors are randomly sampled to form the training set. Here, retrieval performance is evaluated on the NSB and PSB databases for the CM-BOF methods corresponding

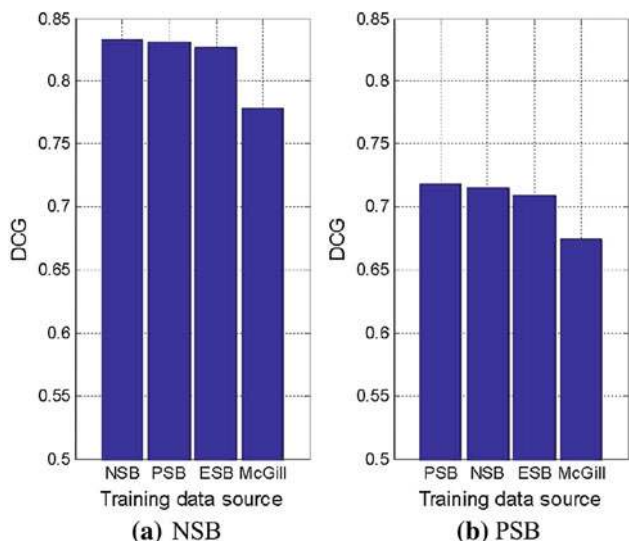


Fig. 14 Influence of the training data source on the retrieval results. **a, b** Compare the DCG values of our CM-BOF methods, corresponding to four different training data sources, run on the NSB and PSB databases, respectively

to four training sets generated from the PSB, NSB, ESB, and McGill databases, respectively. Figure 14 shows the results. Apparently, as we expected, the NSB training data give the best result on the NSB database and the PSB training data give the best result on the PSB database. Moreover, we also observe that better results could be obtained if more similar training data source, compared to the target database, is utilized.

4.4 Influence of the distance function

In the fourth experiment, we compare the performance of our methods, which use different distance functions to calculate the distance between two views.

Here we test four distance functions denoted as D_{MaxHis} , D_{MinHis} , D_{AvgHis} , and D_{L1} , respectively. Assume that view k is described by the word histogram $H_k = \{H_k(j)|j = 0, 1, \dots, N_w - 1\}$, given two histograms H_1, H_2 , the distance between them can be calculated by the following four distance functions, the first three of which are modified from the histogram intersection distance presented in [54].

1. Maximum dissimilarity histogram intersection distance.

$$D_{MaxHis} = 1 - \frac{\sum_{j=0}^{N_w-1} \min(H_1(j), H_2(j))}{\max\left(\sum_{j=0}^{N_w-1} H_1(j), \sum_{j=0}^{N_w-1} H_2(j)\right)}. \tag{16}$$

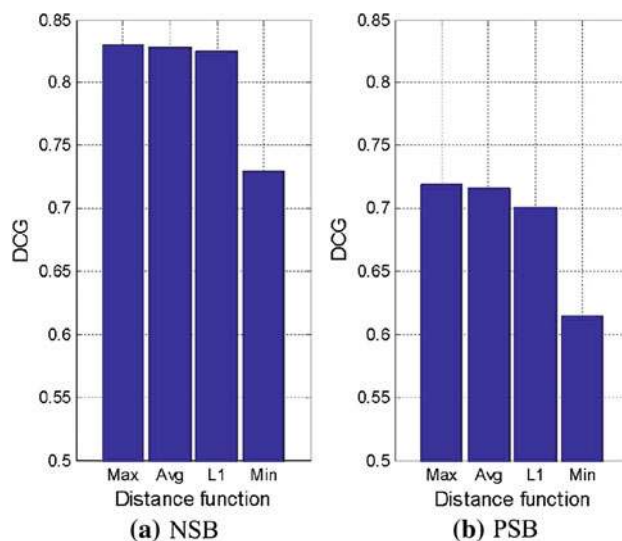


Fig. 15 Influence of the dissimilarity measure on retrieval results. **a, b** Compare the DCG values of our CM-BOF methods, corresponding to four different distance functions, evaluated on two benchmarks. The distance functions D_{MaxHis} , D_{MinHis} , D_{AvgHis} , and D_{L1} are denoted as *Max*, *Min*, *Avg*, and *L1*, respectively

2. Minimum dissimilarity histogram intersection distance.

$$D_{MinHis} = 1 - \frac{\sum_{j=0}^{N_w-1} \min(H_1(j), H_2(j))}{\min\left(\sum_{j=0}^{N_w-1} H_1(j), \sum_{j=0}^{N_w-1} H_2(j)\right)}. \tag{17}$$

3. Average dissimilarity histogram intersection distance.

$$D_{AvgHis} = 1 - \frac{\sum_{j=0}^{N_w-1} \min(H_1(j), H_2(j))}{\left(\sum_{j=0}^{N_w-1} H_1(j) + \sum_{j=0}^{N_w-1} H_2(j)\right)/2}. \tag{18}$$

4. Normalized L1 distance.

$$D_{L1} = \sum_{j=0}^{N_w-1} \left| \frac{H_1(j)}{\sum_{j=0}^{N_w-1} H_1(j)} - \frac{H_2(j)}{\sum_{j=0}^{N_w-1} H_2(j)} \right|. \tag{19}$$

As we can see from Fig. 15, D_{MaxHis} outperforms other three distance functions.

4.5 Comparison of local and global methods

In this section, results of our fifth experiment are presented to discuss the advantages and disadvantages of two methods (CM-BOF and GSMD [30]), which utilize local and global features, respectively, to describe views.

These two methods both capture 66 views and apply the same shape matching scheme, the only difference is that the

local-based method (CM-BOF) uses a word histogram of local features to describe each view while the global-based method (GSMD [30]) represents the view by a global feature vector with 47 elements including 35 Zernike moments, 10 Fourier coefficients, eccentricity and compactness. The comparison is performed using the precision-recall curve on each class of the NSB database. Inspecting the comparison results shown in Fig. 16, we could classify them into the following three categories and suggest several possible reasons. Examples of the first category are displayed in row 1 and 2, where the local-based method significantly outperforms the global-based method. 62.5% objects in the NSB database belong to this category. We speculate that, this is because, for these models in a same class, they have different global appearances but look similar when focusing on local regions, or their local descriptors provide more details than the global features. For instance, cabinet, telephone, biplane, etc., can be better retrieved using the local-based method. Three examples of the second category are shown in row 3, where the global-based method obtains much better results than the local-based one. Only 12.5% models belong to this category. Possible explanations are twofold: on the one hand, local salient features are extracted from unimportant but locally different subparts of these models (e.g., sword's handle); on the other hand, overall appearances of these models (e.g., missile and ant) in the same class are similar but not their local regions. The last row shows the third category, where the local-based and global-based methods get almost the same performance. 25.0% objects, such as sofa, monitor, guitar and so on, belong to this category. To sum up, the local-based method (CM-BOF) is generally superior to the global-base method (GSMD) (the result comparisons for entire databases are shown in Fig. 17), however, the global-base method may outperform the local-based method when searching for certain kinds of models. Furthermore, these two methods represent a depth-buffer view in quite different manners. Therefore, to some extent, they are complementary and it is possible to create a more discriminative descriptor using the combination of local feature and global feature to represent the depth-buffer views.

4.6 Comparison with the state of the art

In this section, we compare the performance of our algorithms with the state of the art in applications of rigid and non-rigid 3D shape retrieval, respectively.

4.6.1 Application in rigid 3D shape retrieval

For the application of rigid 3D shape retrieval, experiments are run on the NSB and PSB databases, respectively. Figure 17 shows the precision-recall curves on afore-

mentioned two benchmarks for eight methods listed as follows:

- *D2* A method describing a 3D object as a histogram of distances between pairs of randomly sampled points on the surface [38]. The number of histogram bins is chosen as 64.
- *SHD* A method describing a 3D object as a feature vector consisting of spherical harmonic coefficients extracted from three spherical functions giving the maximal distance from center of mass as a function of spherical angle. See [23, 30] for details.
- *LFD* A well-known visual similarity-based method proposed in [10]. See Sect. 2 for details. The length of the feature vector is 4,700.
- *GSMD* See Sect. 4.5 for details. The number of view-points is selected as 66.
- *BF-SIFT* A method representing a 3D object as a single word histogram by vector quantizing the visual salient SIFT descriptors [37]. Here, the depth-buffer views are captured on the vertices of a unit geodesic sphere. The number of views is selected as 66 and the length of the feature vector is 1,500.
- *PANORAMA* A recently proposed method [41] that describes a 3D object using 2D fourier coefficients and 2D wavelet coefficients extracted from its panoramic views. Here, we directly use the original implementation developed by the authors [41].
- *CM-BOF* See Sect. 3 for details. The default settings are chosen here and thus the average length of the feature vector is 3,320.
- *Hybrid (CM-BOF+GSMD)* A composite method based on a linear combination of CM-BOF and GSMD. More specifically, in this method, a view is expressed by a feature vector consisting of two different kinds of shape descriptors, which are used in CM-BOF and GSMD, with pre-specified weights. We experimentally select the weights as $W_{\text{local}} = 7.0$ and $W_{\text{global}} = 1.0$ for local and global features, respectively, by maximizing the retrieval accuracy on the PSB train set with base classification. The shape matching scheme and other parameters are exactly the same as the CM-BOF algorithm described above.

As we can see from Fig. 17, for the PSB database, the Hybrid method clearly outperforms other seven methods, among which PANORAMA and CM-BOF take the second place and the third place, respectively. While, for the NSB database, our Hybrid method performs slightly worse than the PANORAMA approach. This is probably due to the less diversity in models of the NSB database compared to PSB. Using four quantitative measures [i.e., Nearest neighbor (1-NN), First-tier (1-Tier), second-tier (2-Tier), and discounted cumulative gain (DCG)] we also

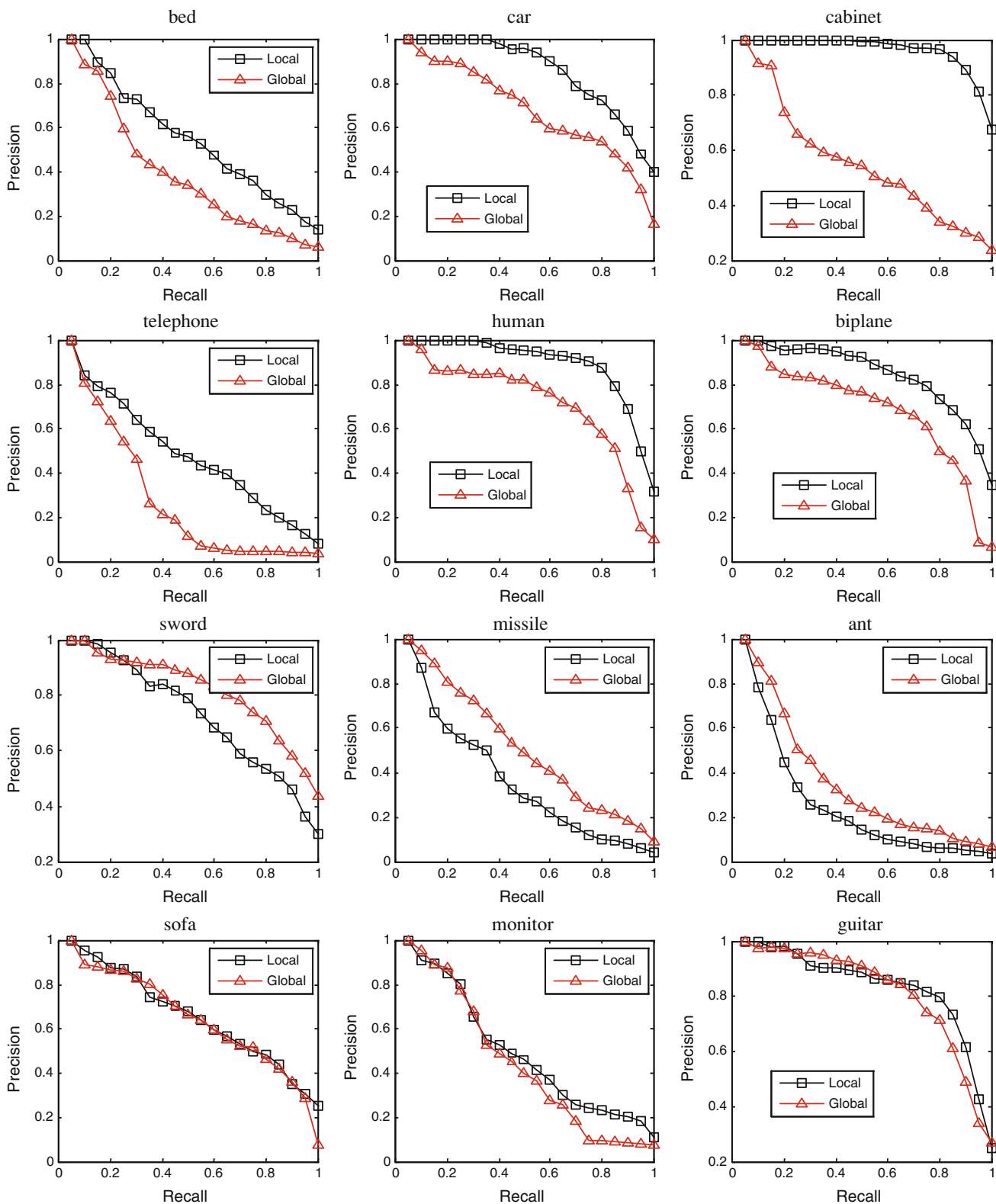


Fig. 16 Precision-recall curves for specific categories on the NSB database. In this figure, *Local* denotes the local feature-based method (CM-BOF), while the global feature-based method (GSMD) is denoted as *Global*

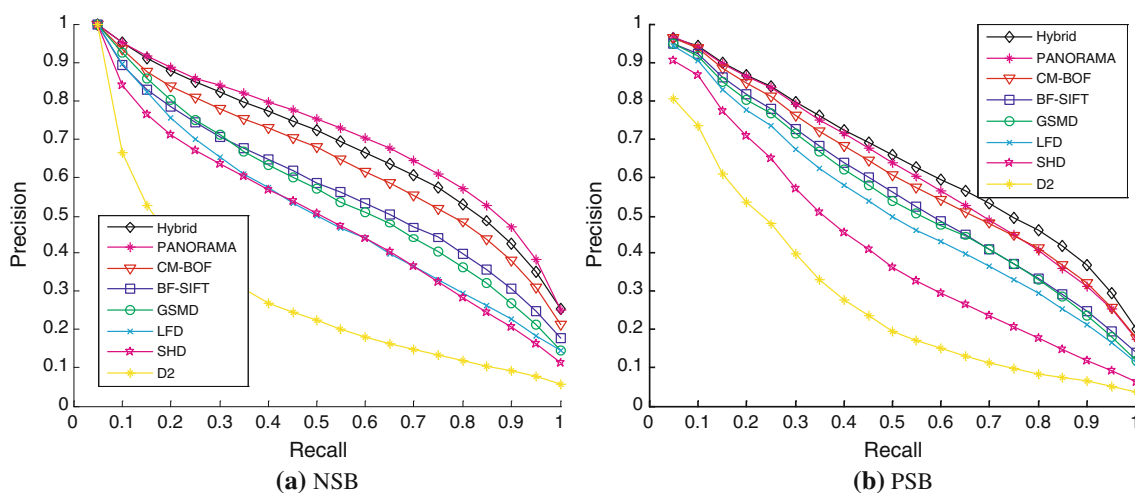


Fig. 17 Precision-recall curves of our methods (i.e., Hybrid and CM-BOF) and other six approaches run on the two rigid 3D shape benchmarks. **a, b** Shows the results evaluated on the NSB and PSB databases, respectively

Table 2 Comparing retrieval results of our methods (first two rows) with other state-of-the-art approaches on the PSB test set with base classification

	1-NN (%)	1-Tier (%)	2-Tier (%)	DCG (%)
Hybrid	75.4	50.9	64.0	74.6
CM-BOF	73.1	47.0	59.8	72.0
PANORAMA	75.3	47.9	60.3	—
GSMD+SHD+R	73.1	47.2	60.2	72.1
MDLA-DPD	68.8	43.6	54.2	67.8
DESIRE	66.5	40.3	51.2	66.3
LFD	65.7	38.0	48.7	64.3

compare our Hybrid (CM-BOF+GSMD) and CM-BOF methods quantitatively with state-of-the-art approaches including PANORAMA [41], MDLA-DPD [9], GSMD+SHD+R [30], DESIRE [59], and LFD [10] on the PSB database. As shown in Table 2, our Hybrid method significantly outperforms all other methods compared here, while the CM-BOF algorithm, whose feature vector is only of dimension 3,320 on average, also obtains superior or comparable 3D shape retrieval performance. Moreover, for our CM-BOF method, comparing a pair of 3D objects takes less than 1.0 millisecond and, with the GPU-based implementation [62], the feature extraction of an object can be finished within 5.0 s.

Furthermore, we also run our methods (i.e., Hybrid and CM-BOF) on the GWSB database and compare them with other eight approaches, all of which have been evaluated in a contest named “the SHREC’10 Track: Generic 3D Warehouse [43]”. Figure 18 again demonstrates the superiority of our methods compared to the state of the art in rigid 3D shape retrieval. Finally, it is also worth mentioning that our rigid shape retrieval methods are able to effectively handle any kinds of polygon meshes, even those polygon soups consisting of unorganized triangles.

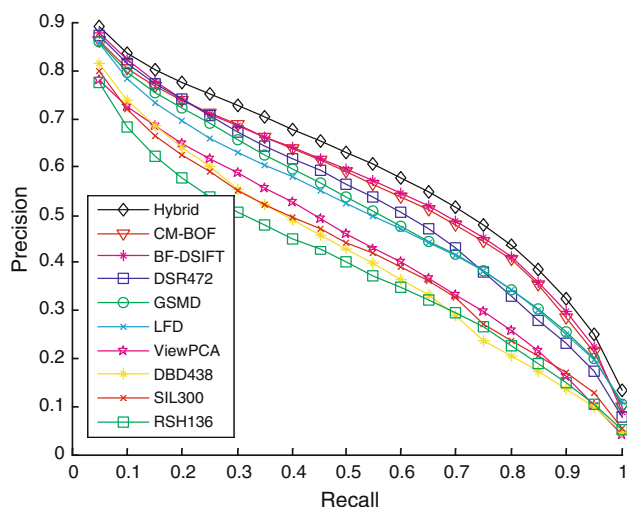


Fig. 18 Precision-recall curves of our methods (i.e., Hybrid and CM-BOF) and other eight approaches run on the GWSB database

4.6.2 Application in non-rigid 3D shape retrieval

For the application of non-rigid 3D shape retrieval, experiments are run on the McGill and SHREC’11 Non-rigid

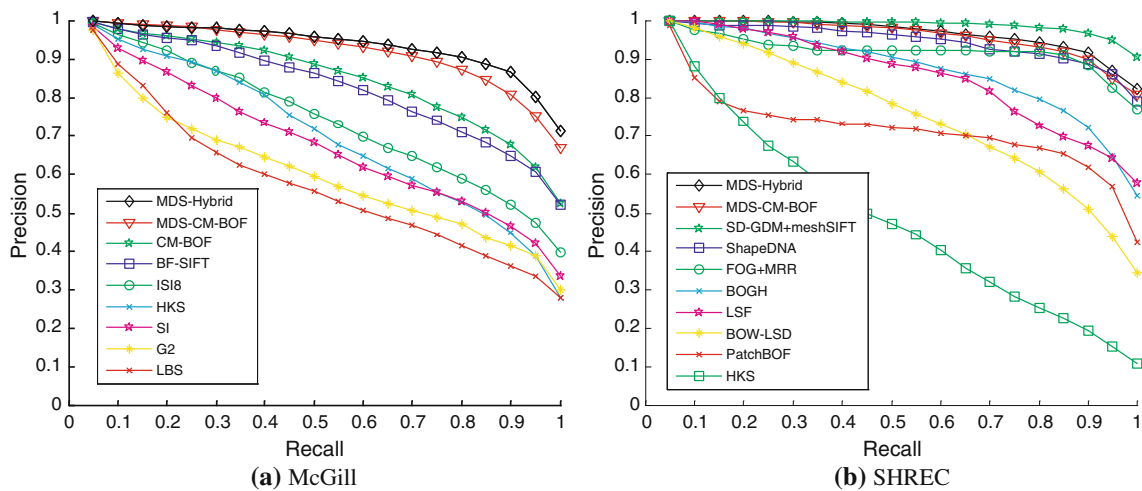


Fig. 19 Precision-recall curves of our methods (i.e., MDS-Hybrid and MDS-CM-BOF) and other existing approaches run on the two non-rigid 3D shape benchmarks. **a, b** Shows the results evaluated on the McGill and SHREC’11 Non-rigid 3D Shape databases, respectively

3D Shape databases, respectively. Experimental results on the McGill database are shown in Fig. 19a to compare the retrieval performance of the proposed methods (i.e., MDS-Hybrid, MDS-CM-BOF, and CM-BOF) with the following 6 non-rigid 3D shape retrieval algorithms: BF-SIFT [37], intrinsic spin images (ISI8) [61], heat kernel signatures (HKS) [6], spin images (SI) [22], the shape distribution of Geodesic distance (G2) [34], and Laplace–Beltrami spectrum (LBS) [45]. As mentioned above, the Hybrid method, which represents 2D views using both local and global features, is basically a combination of the CM-BOF and GSMD methods. For convenience, the 3D canonical form obtained by applying Least Squares MDS with the SMACOF algorithm is denoted as MDS. Therefore, MDS-Hybrid and MDS-CM-BOF stand for the retrieval methods using the Hybrid and CM-BOF approaches, respectively, with MDS canonical forms. As we can see from Fig. 19a, our methods with MDS embedding (i.e., MDS-Hybrid and MDS-CM-BOF) could perform markedly better than all others approaches. This is mainly due to the utilization of both 3D canonical forms and salient local features. Moreover, our MDS-Hybrid, MDS-CM-BOF methods are also compared quantitatively with several state-of-the-art approaches for non-rigid 3D retrieval

including EMD-PPPT [1], EMD-MPEG7 [1], BF-SIFT [37] and ISI8 [61]. As it can be observed from Table 3, our methods obtain significantly better performance compared to the state of the art.

Finally, we also evaluate our methods (i.e., MDS-Hybrid and MDS-CM-BOF) on the SHREC’11 Non-rigid 3D Shape database and compare the performance of our methods with other eight approaches that have been tested in the SHREC’11 Track: Shape Retrieval on Non-rigid 3D Watertight Meshes [27]. As we can see from Fig. 19b and Table 4, our methods perform only slightly worse than the best one (i.e., SD-GDM+meshSIFT), which employs the combination of a global spectral 3D shape descriptor and a salient 3D local feature. Moreover, our methods also give 100% for Nearest Neighbor (i.e., 1-NN). At last, we show some examples of queries and their corresponding top 6 retrieved models from the SHREC’11 Non-rigid 3D Shape database using our MDS-CM-BOF algorithm in Fig. 20. It can be observed that the retrieved 3D models in the top 6 positions of the rank lists all belong to the same categories of their corresponding queries, which again validates the effectiveness of our methods in the application of non-rigid 3D shape retrieval. Nevertheless, it should be pointed out that the proposed non-rigid

Table 3 Comparing retrieval results of our methods (first two rows) with the state of the art on the McGill articulated 3D shape database

	NN (%)	1-Tier (%)	2-Tier (%)	DCG (%)
MDS-Hybrid	99.2	87.6	97.2	97.7
MDS-CM-BOF	99.2	84.8	96.2	97.4
EMD-PPPT	97.6	74.1	91.1	93.3
EMD-MPEG7	93.3	69.2	88.9	90.8
BF-SIFT	97.3	74.6	87.0	93.7
ISI8	95.3	64.2	79.9	90.0

Table 4 Retrieval performance of our methods (first two rows) and other eight approaches evaluated using four standard measures on the SHREC'11 Non-rigid 3D Shape database

Methods	1-NN (%)	1-Tier (%)	2-Tier (%)	DCG (%)
MDS-Hybrid	100.0	93.2	97.5	98.9
MDS-CM-BOF	100.0	92.2	97.1	98.6
SD-GDM+meshSIFT	100.0	97.2	99.0	99.6
ShapeDNA	99.2	91.5	95.7	97.8
FOG+MRR	96.0	88.1	94.6	95.9
BOGH	99.3	81.1	88.4	94.9
LSF	99.5	79.9	86.3	94.3
BOW-LSD	95.5	67.2	80.3	89.7
PatchBOF	74.8	64.2	83.3	83.7
HKS	83.7	40.6	49.7	73.0

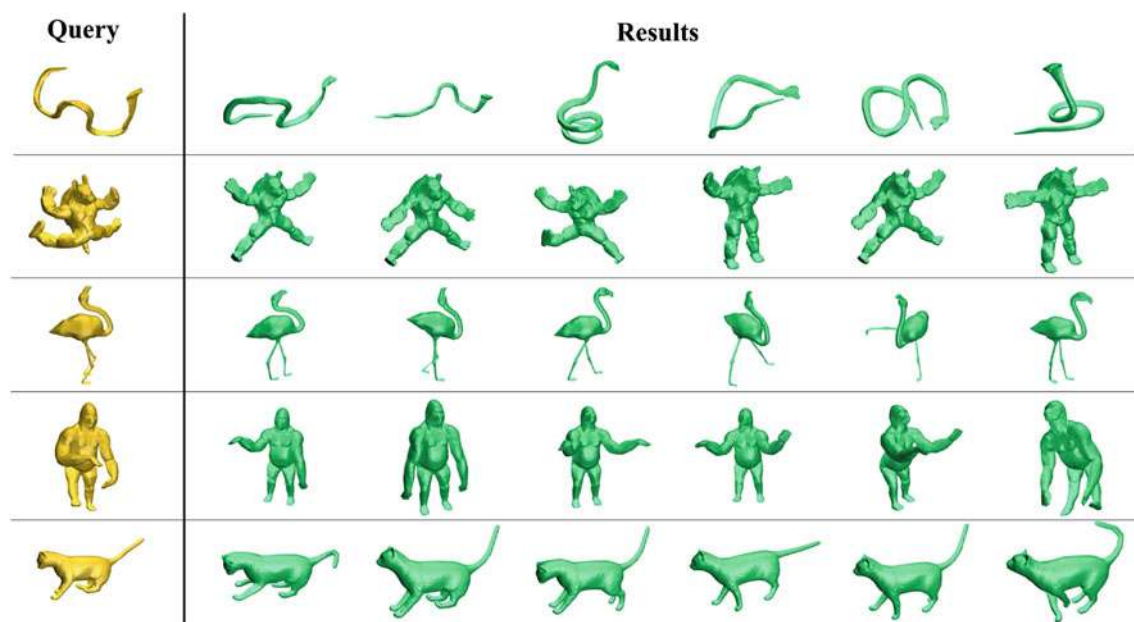


Fig. 20 Examples of queries (*first column*) from the SHREC'11 Non-rigid 3D Shape database and the corresponding top 6 retrieved models using our MDS-CM-BOF method. The retrieved models are ranked from *left to right* based on the increasing order of dissimilarity

shape retrieval algorithms can only perform well for databases consisting of well-constructed polygon meshes but not polygon soups, because we need to calculate the geodesic distance between each pair of points sampled on the surface when implementing MDS embedding.

5 Conclusion

In this paper, we presented a novel visual similarity-based 3D shape retrieval method (CM-BOF) using Clock Matching and Bag-of-Features. The key idea is to describe each

depth-buffer view captured around the 3D model as a word histogram, which is obtained by the vector quantization of the view's salient local features, and employ a multi-view shape matching to calculate the dissimilarity between two objects. When applying the CM-BOF method to retrieve non-rigid 3D models, MDS embedding should be utilized before pose normalization to calculate the canonical form for each object. We also carried out a set of experiments to investigate several critical issues of our CM-BOF method, including the impact of the number of views, codebook, training data, and distance function on the performance of 3D shape retrieval. It can be seen that clustering is not necessary for the method, and our local feature-based method is somehow complemen-

tary with respect to the global feature-based method (GSMD [30]). The experimental results also demonstrated that our methods are superior or comparable to the state of the art in applications of both rigid and non-rigid 3D shape retrieval.

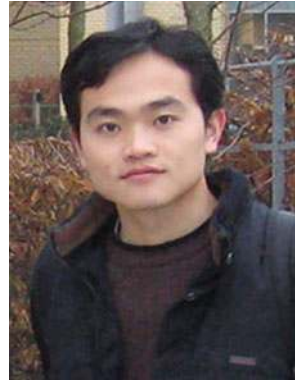
Acknowledgments This work has been supported by National Natural Science Foundation of China (Grant No. 61202230), China Postdoctoral Science Foundation (Grant No.: 2012M510274), the SIMA program and the Shape Metrology IMS.

References

- Agathos, A.: Part based 3D representation for the retrieval of 3D graphical models. PhD thesis, the University of Athens (2009)
- Aim@Shape: SHape REtrieval Contest (SHREC). <http://www.aimatshape.net/event/SHREC/> (2006)
- Ankerst, M., Kastenmuller, G., Kriegel, H., Seidl, T.: Nearest neighbor classification in 3D protein databases. In: Proceedings of the Seventh International Conference on Intelligent Systems for Molecular Biology, pp. 34–43 (1999)
- Bespalov, D., Regli, W., Shokoufandeh, A.: Reeb graph based shape retrieval for cad. In: Proceedings of the ASME Design Engineering Technical Conferences, Computers and Information in Engineering Conference (2003)
- Borg, I., Groenen, P.: Modern Multidimensional Scaling—Theory and Applications. Springer, Berlin (1997)
- Bronstein, A.M., Bronstein, M.M., Guibas, L.J., Ovsjanikov, M.: Shape google: geometric words and expressions for invariant shape retrieval. *ACM Trans. Graph.* **30**(1), 1–20 (2011)
- Bronstein, A.M., Bronstein, M.M., Kimmel, R.: Numerical Geometry of Non-Rigid Shapes. Springer, Berlin (2008)
- Chaouch, M., Verroust-Blondet, A.: Enhanced 2D/3D approaches based on relevance index for 3D-shape retrieval. In: Proceedings of the SMI'06, pp. 36–36 (2006)
- Chaouch, M., Verroust-Blondet, A.: A new descriptor for 2D depth image indexing and 3D model retrieval. In: Proceedings of the ICIP'07, vol. 6, pp. 373–376 (2007)
- Chen, D.Y., Tian, X.P., Shen, Y.T., Ouhyoung, M.: On visual similarity based 3D model retrieval. In: Proceedings of the Eurographics 2003, pp. 223–232 (2003)
- Daras, P., Axenopoulos, A.: A 3D shape retrieval framework supporting multimodal queries. *IJCV* **89**(2–3), 229–247 (2010)
- Daras, P., Axenopoulos, A., Litos, G.: Investigating the effects of multiple factors towards more accurate 3D object retrieval. *IEEE Trans. Multimed.* **14**(2), 374–388 (2012)
- Elad, A., Kimmel, R.: On bending invariant signatures for surface. *IEEE Trans. PAMI* **25**(10), 1285–1295 (2003)
- Fang, R., Godill, A., Li, X., Wagan, A.: A new shape benchmark for 3D object retrieval. In: Proceedings of the ISVC'08, pp. 381–392 (2008)
- Fei-Fei, L., Perona, P.: A bayesian hierarchical model for learning natural scene categories. In: Proceedings of the CVPR'05, pp. 524–531 (2005)
- Frome, A., Huber, D., Kolluri, R., Bulow, T., Malik, J.: Recognizing objects in range data using regional point descriptors. In: Proceedings of the ECCV'04 (2004)
- Funkhouser, T., Shilane, P.: Partial matching of 3D shapes with priority-driven search. In: Proceedings of the SGP'06, pp. 131–142 (2006)
- Furuya, T., Ohbuchi, R.: Dense sampling and fast encoding for 3D model retrieval using bag-of-visual features. In: Proceedings of the CIVR'09 (2009)
- Gal, R., Cohen-Or, D.: Salient geometric features for partial shape matching and similarity. *ACM Trans. Graph.* **25**(1), 130–150 (2006)
- Jayanti, S., Kalyanaraman, Y., Iyer, N., Ramani, K.: Developing an engineering shape benchmark for CAD models. *Comput. Aided Des.* **38**(9), 939–953 (2006)
- Jegou, H., Douze, M., Schmid, C.: Improving bag-of-features for large scale image search. *IJCV* **87**(3), 316–336 (2010)
- Johnson, A.E., Hebert, M.: Using spin images for efficient object recognition in cluttered 3D scenes. *IEEE Trans. PAMI* **21**(5), 433–449 (1999)
- Kazhdan, M., Funkhouser, T., Rusinkiewicz, S.: Rotation invariant spherical harmonic representation of 3D shape descriptors. In: Proceedings of the SGP'03, vol. 43, pp. 156–164 (2003)
- Laga, H., Takahashi, H., Nakajima, M.: Spherical wavelet descriptors for content-based 3D model retrieval. In: Proceedings of the SMI'06, pp. 15–15 (2006)
- Li, X., Godill, A., Wagan, A.: Spatially enhanced bags of words for 3D shape retrieval. In: Proceedings of the ISVC'08, pp. 349–358 (2008)
- Lian, Z., Godil, A., Bustos, B., Daoudi, M., Hermans, J., Kawamura, S., Kurita, Y., Lavoue, G., Nguyen, H., Ohbuchi, R., Ohkita, Y., Ohishi, Y., Porikli, F., Reuter, M., Sipiran, I., Smeets, D., Suetens, P., Tabia, H., Vandermeulen, D.: A comparison of methods for non-rigid 3D shape retrieval. *Pattern Recognit.* **46**(1), 449–461 (2013)
- Lian, Z., Godil, A., Bustos, B., Daoudi, M., Hermans, J., Kawamura, S., Kurita, Y., Lavoue, G., Nguyen, H.V., Ohbuchi, R., Ohkita, Y., Ohishi, Y., Porikli, F., Reuter, M., Sipiran, I., Smeets, D., Suetens, P., Tabia, H., Vandermeulen, D.: SHREC'11 track: shape retrieval on non-rigid 3D watertight meshes. In: Proceedings of the 3DOR'11, pp. 79–88 (2011)
- Lian, Z., Godil, A., Sun, X.: Visual similarity based 3D shape retrieval using bag-of-features. In: Proceedings of the SMI'10, pp. 25–36 (2010)
- Lian, Z., Godil, A., Sun, X., Zhang, H.: Non-rigid 3D shape retrieval using multidimensional scaling and bag-of-features. In: Proceedings of the International Conference on Image Processing (ICIP 2010), pp. 3181–3184 (2010)
- Lian, Z., Rosin, P.L., Sun, X.: Rectilinearity of 3D meshes. *IJCV* **89**(2–3), 130–151 (2010)
- Liu, Y., Zha, H., Qin, H.: Shape topics: A compact representation and new algorithms for 3D partial shape retrieval. In: Proceedings of the CVPR'06, pp. 2025–2032 (2006)
- Lo, T.R., Siebert, J.P.: Local feature extraction and matching on range images: 2.5D SIFT. *Comput. Vis. Image Underst.* **113**(12), 1235–1250 (2009)
- Lowe, D.G.: Distinctive image features from scale-invariant keypoints. *IJCV* **60**(2), 91–110 (2004)
- Mahmoudi, M., Sapiro, G.: Three-dimensional point cloud recognition via distributions of geometric distances. *Graph. Models* **71**(1), 22–31 (2009)
- Mikolajczyk, K., Schmid, C.: A performance evaluation of local descriptors. *IEEE Trans. PAMI* **27**(10), 1615–1630 (2005)
- Nowak, E., Jurie, F., Triggs, B.: Sampling strategies for bag-of-features image classification. In: Proceedings of the ECCV'06, pp. 490–503 (2006)
- Ohbuchi, R., Osada, K., Furuya, T., Banno, T.: Salient local visual features for shape-based 3D model retrieval. In: Proceedings of the SMI'08, pp. 93–102 (2008)
- Osada, R., Funkhouser, T., Chazelle, B., Dobkin, D.: Shape distributions. *ACM Trans. Graph.* **21**(4), 807–832 (2002)
- Papadakis, P.: Content-based 3D model retrieval considering the user's relevance feedback. PhD thesis, the University of Athens (2009)

40. Papadakis, P., Pratikakis, I., Perantonis, S., Theoharis, T.: Efficient 3d shape matching and retrieval using a concrete radialized spherical projection representation. *Pattern Recognit.* **40**(9), 2437–2452 (2007)
41. Papadakis, P., Pratikakis, I., Theoharis, T., Perantonis, S.: PANORAMA: A 3D shape descriptor based on panoramic views for unsupervised 3D object retrieval. *IJCV* **89**(2–3), 177–192 (2010)
42. Passalis, G., Theoharis, T., Kakadiaris, I.A.: PTK: A novel depth buffer-based shape descriptor for three-dimensional object retrieval. *Vis. Comput.* **23**(1), 5–14 (2007)
43. Porethi, V., Godill, A., Dutagaci, H., Furuya, T., Lian, Z., Ohbuchi, R.: SHREC'10 track: generic 3D warehouse. In: *Proceedings of the 3DOR'10*, pp. 93–100 (2010)
44. Pu, J., Lou, K., Ramani, K.: A 2D sketch-based user interface for 3D CAD model retrieval. *Comput. Aided Des. Appl.* **2**(6), 717–725 (2005)
45. Reuter, M., Wolter, F.E., Peinecke, N.: Laplace-spectra as fingerprints for shape matching. In: *Proceedings of the SPM'05*, pp. 101–106 (2005)
46. Ruggeri, M.R., Patane, G., Spagnuolo, M., Saupe, D.: Spectral-driven isometry-invariant matching of 3D shapes. *IJCV* **89**(2–3), 248–265 (2010)
47. Shih, J., Hsing, C., Wang, J.: A new 3D model retrieval approach based on the elevation descriptor. *Pattern Recognit.* **40**(1), 283–295 (2007)
48. Shilane, P., Funkhouser, T.: Distinctive regions of 3D surfaces. *ACM Trans. Graph.* **26**(2) (2007)
49. Shilane, P., Min, P., Kazhdan, M., Funkhouser, T.: The princeton shape benchmark. In: *Proceedings of the SMI'04*, pp. 167–178 (2004)
50. Siddiqi, K., Zhang, J., Macrini, D., Shokoufandeh, A., Bouix, S., Dickinson, S.: Retrieving articulated 3D models using medial surfaces. *Mach. Vis. Appl.* **19**(4), 261–275 (2008)
51. Sivic, J., Zisserman, A.: Video google: a text retrieval approach to object matching in videos. In: *Proceedings of the ICCV'03*, pp. 1470–1477 (2003)
52. Smeets, D., Hermans, J., Vandermeulen, D., Suetens, P.: Isometric deformation invariant 3d shape recognition. *Pattern Recognit.* **45**(7), 2817–2831 (2012)
53. Sundar, H., Silver, D., Gavani, N., Dickinson, S.: Skeleton based shape matching and retrieval. In: *Proceedings of the SMI'03*, pp. 130–139 (2003)
54. Swain, M.J., Ballard, D.H.: Color indexing. *IJCV* **7**(1), 11–32 (1991)
55. Tal, A., Zuckerberger, E.: Mesh retrieval by components. *Adv. Comput. Graph. Comput. Vis.* 44–57 (2007)
56. Tangelder, J.W., Veltkamp, R.C.: A survey of content based 3D shape retrieval methods. *Multimed. Tools Appl.* **39**(3), 441–471 (2008)
57. Toldo, R., Castellani, U., Fusiello, A.: Visual vocabulary signature for 3D object retrieval and partial matching. In: *Proceedings of the 3DOR'08*, pp. 21–28 (2009)
58. Vedaldi, A., Fulkerson, B.: VLFeat: an open and portable library of computer vision algorithms. <http://www.vlfeat.org/> (2008)
59. Vranić, D.V.: DESIRE: a composite 3D-shape descriptor. In: *Proceedings of the ICME'05* (2005)
60. Vranić, D.V., Saupe, D., Richter, J.: Tools for 3D-object retrieval: Karhunen-loeve transform and spherical harmonics. In: *Proceedings of the 2001 IEEE Fourth Workshop on Multimedia Signal Processing*, pp. 293–298 (2001)
61. Wang, X., Liu, Y., Zha, H.: Intrinsic spin images: a subspace decomposition approach to understanding 3D deformable shapes. In: *Proceedings of the 3DPVT'10*, pp. 17–20 (2010)
62. Wu, C.: SiftGPU: a GPU implementation of David Lowe's SIFT. <http://cs.unc.edu/ccwu/siftgpu/> (2009)

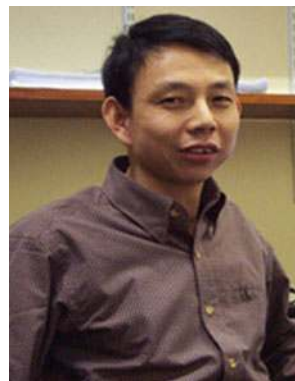
Author Biographies



Zhouhui Lian received the Ph.D. degree from Beihang University, China in 2011. He worked as a guest researcher at NIST, Gaithersburg, USA, from 2009 to 2011. He is currently a postdoctoral researcher at the Institute of Computer Science and Technology, Peking University, China. His main research interests include computer graphics and computer vision.



Afzal Godil received the master degree from the University of Arizona, USA. He is currently a project leader in the Information Technology Laboratory, National Institute of Standards and Technology (NIST), Gaithersburg, USA. His main research interests include shape retrieval, shape analysis, digital human modeling and 3D visualization.



Xianfang Sun received the Ph.D. degree from the Institute of Automation, Chinese Academy of Sciences in 1994. He is currently a lecturer at the School of Computer Science, Cardiff University, Wales, UK. His main research interests include computer vision, computer graphics, pattern recognition and artificial intelligence.



Jianguo Xiao received the master degree from Peking University, China in 1989. He is currently a professor and the director at the Institute of Computer Science and Technology, Peking University, China. His main research interests include computer graphics, image and video processing, web information processing and text mining.