Int J Comput Vis (2013) 102:221–238
DOI 10.1007/s11263-012-0548-1

Feature-Preserved 3D Canonical Form

Zhouhui Lian · Afzal Godil · Jianguo Xiao

Received: 4 November 2011 / Accepted: 16 July 2012 / Published online: 28 July 2012

Abstract Measuring the dissimilarity between non-rigid objects is a challenging problem in 3D shape retrieval. One potential solution is to construct the models' 3D canonical forms (i.e., isometry-invariant representations in 3D Euclidean space) on which any rigid shape matching algorithm can be applied. However, existing methods, which are typically based on embedding procedures, result in greatly distorted canonical forms, and thus could not provide satisfactory performance to distinguish non-rigid models.

In this paper, we present a feature-preserved canonical form for non-rigid 3D watertight meshes. The basic idea is to naturally deform original models against corresponding initial canonical forms calculated by Multidimensional Scaling (MDS). Specifically, objects are first segmented into near-rigid subparts, and then, through properly-designed rotations and translations, original subparts are transformed into poses that correspond well with their positions and directions on MDS canonical forms. Final results are obtained by solving nonlinear minimization problems for optimal alignments and smoothing boundaries between subparts. Experiments on two non-rigid 3D shape benchmarks not only clearly verify the advantages of our algorithm against existing approaches, but also demonstrate that, with the help of the proposed canonical form, we can obtain significantly better retrieval accuracy compared to the state of the art.

Keywords Canonical form · Multidimensional scaling · 3D shape retrieval · Non-rigid

1 Introduction

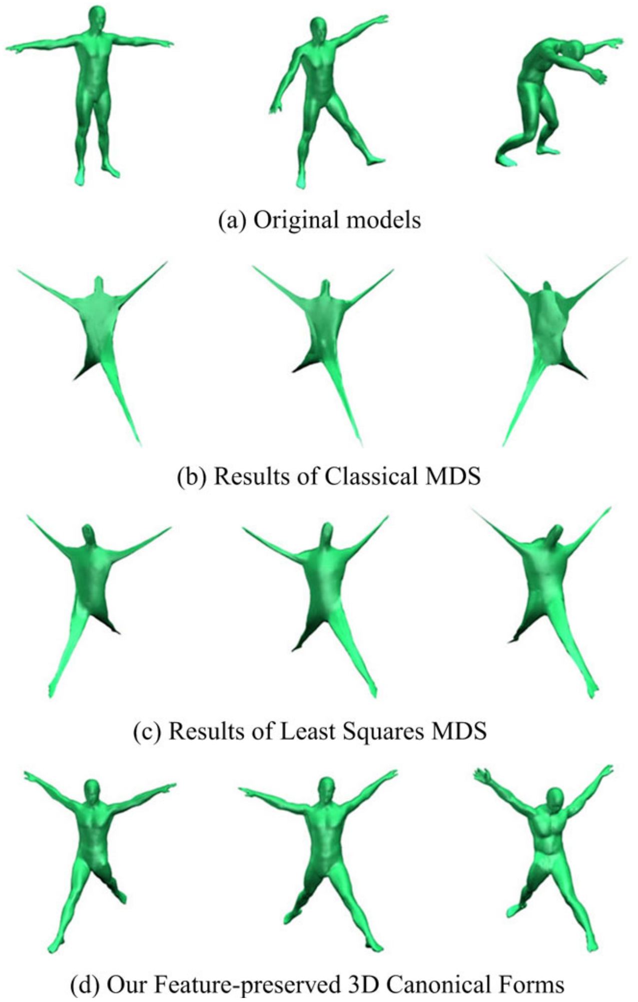
With the ever increasing accumulation of 3D models, how to accurately and efficiently retrieve these data has become an important problem in computer vision, pattern recognition, computer graphics, mechanic CAD, and many other fields (Shilane et al. 2004; Tangelder and Veltkamp 2008). One of most challenging issues in this problem is the calculation of dissimilarity between non-rigid objects that are commonly seen in our surroundings. Take Fig. 1(a) for an example, a man appears in three distinctive poses, but these models represent the same object in despite of having different appearances. In order to compare non-rigid 3D models quickly and effectively, it is often desired that the shapes can be represented by some discriminative signatures which are invariant or approximately invariant under various isometric transformations (i.e., rigid-body transformations, non-rigid bending and articulation).

While a large number of retrieval methods for rigid 3D shapes have been proposed in the last few years, there has been considerably less work for non-rigid models. In general, existing non-rigid 3D shape retrieval methods can be roughly classified into algorithms using local features, topological structures, isometry-invariant global geometric properties, direct shape matching, or canonical forms. Although these algorithms are all guaranteed to be isometry-invariant, they are still not well suited for practical applications in

Z. Lian (✉) · J. Xiao
Institute of Computer Science and Technology, Peking University, Beijing, P.R. 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


(a) Original models

(b) Results of Classical MDS

(c) Results of Least Squares MDS

(d) Our Feature-preserved 3D Canonical Forms

Fig. 1 Non-rigid models (**a**) and their 3D canonical forms obtained using Classical MDS (**b**), Least Squares MDS (**c**) and the proposed method (**d**), respectively

non-rigid 3D shape retrieval. This is mainly due to the fact that typically they are either computationally expensive or poor in discrimination. Further discussions are provided in Sect. 2. Perhaps, the utilization of canonical forms (unless otherwise specified, *canonical form* mentioned in this paper means the canonical form in 3D Euclidean space) is potentially the most effective way to address the problem of non-rigid 3D shape matching. As we know, through the calculation of canonical forms, deformable models can be normalized into particular 3D representations which are unique and isometry-invariant. Then, any shape retrieval approach, even methods specifically designed for rigid models, can be applied to measure the similarity between non-rigid models. For instance, the visual similarity based method (Chen et al. 2003; Lian et al. 2010c), which has been widely acknowledged as the most powerful and practical approach for rigid 3D shape retrieval (Shilane et al. 2004), is essentially unsuitable for the shape matching of non-rigid objects. This is because, when a 3D model is articulated or bent, serious occlusions may occur and numerous noises could be generated in the views captured around the object. Owing to the introduction of canonical forms, the tough problem of non-rigid shape matching is converted into a simpler and well-studied rigid shape matching problem. Ideally, state-of-the-art approaches including many view-based methods can then be utilized to achieve excellent performance for non-rigid 3D shape retrieval.

However, existing methods that are typically based on embedding procedures could inevitably result in canonical forms with serious distortions (see Figs. 1(b) and (c)). Since the concept of 3D canonical forms was first proposed by Elad and Kimmel in 2003 (Elad and Kimmel 2003), no more progress has been made for the improvement of the qualities of their canonical forms. To the best of our knowledge, our work (Lian and Godil 2011) is the first to obtain feature-preserved 3D canonical forms, and this article is the extended version of the conference paper. Up to now, the Least Squares MDS method employed in Elad and Kimmel (2003) is still considered to be the best way to construct 3D canonical forms with least distortions. Examples of such embedding results are demonstrated in Fig. 1(c). As we can see, compared to original models, important features like hands, feet, and heads are significantly distorted on their canonical forms. It is reasonable to infer that, based on these kinds of canonical forms, objects with similar topology but varied details could not be well distinguished. That is the major reason why previous methods using 3D canonical forms could not obtain satisfactory retrieval performance.

In this paper, a feature-preserved canonical form is proposed for non-rigid 3D watertight meshes. The basic idea is to consider MDS embedding results as references and then naturally deform the original meshes against them. In this manner, our new canonical forms not only have the isometry-invariant property but also preserve important details on the original surfaces (see Fig. 1(d) for some examples). Specifically, features that our method preserves for 3D canonical forms are local structures and surface details which remain unchanged under isometric transformations. To achieve this goal, 3D meshes are first automatically segmented into near-rigid subparts using a new approach. Afterwards, we translate and rotate these segmented subparts into new positions and directions that can be matched well with their corresponding subparts on MDS canonical forms. Finally, we obtain our feature-preserved results by solving several energy minimization problems for optimal assembling and smoothing boundaries between subparts.

The main contribution of this paper is the novel idea of creating feature-preserved canonical forms from MDS embedding results in 3D Euclidean space. We provide an intuitive framework to achieve this goal and demonstrate the advantages of our method against existing canonical forms by several experiments conducted on the commonly-used McGill articulated 3D shape benchmark (Siddiqi et al.

2008) and a specifically-designed database. We also find that, via the utilization of our canonical forms, some rigid shape matching algorithms can obtain markedly better performance, in term of searching accuracy, than other non-rigid 3D shape retrieval methods in the literature.

The rest of this paper is organized as follows. Section 2 reviews related work. Section 3 describes the mathematical background of Multidimensional Scaling. Section 4 presents an explicit description of our method. Results of the feature-preserved canonical form are shown and compared in Sect. 5, which also demonstrates the application of our method in non-rigid 3D shape retrieval. Finally, we discuss and conclude the paper in Sect. 6.

2 Related Work

Shape-based 3D object retrieval, concentrating on the representation and comparison of 3D models based on their intrinsic shapes, has been extensively studied in recent years. Until today, a large amount of 3D shape retrieval methods have been proposed, including Shape Distribution (D2) Osada et al. (2002), Spherical Harmonic Descriptor (SHD) Kazhdan et al. (2003), Light Field Descriptor (LFD) Chen et al. (2003), etc. However, most of these methods were specifically designed for rigid models, and how to effectively and efficiently compare non-rigid 3D models is still considered to be a challenging problem. For more details, we refer the reader to some good surveys (Tangelder and Veltkamp 2008; Shilane et al. 2004).

One popular approach for non-rigid 3D shape retrieval is to compare models based on their local features, which are robust against isometric transformations. For example, Liu et al. (2006) made use of the well-known Spin Images (Johnson and Hebert 1999), and represented a 3D object as a word histogram by vector quantizing all local features extracted from the model. Funkhouser and Shilane (2006) selected distinctive multi-scale local features yielded by applying Spherical Harmonic transformation on salient local regions, and then applied a Priority-Driven search to achieve partial matching. Gal et al. (2007) introduced a pose-oblivious shape descriptor that is actually a 2D histogram combining the distributions of Euclidean distances in local regions and the distributions of geodesic distances for the whole object. Ovsjanikov et al. (2009) employed the Heat Kernel Signature (HKS) (Sun et al. 2009), which is based on the properties of the heat diffusion process on a 3D shape, and a spatially-sensitive bags of features approach to solve the problem of searching non-rigid models in large databases. Ohbuchi et al. (2008) presented a view-based method using salient local features (SIFT Lowe 2004). They represented a 3D model as a word histogram by using bag-of-features for salient local descriptors extracted on the depth-buffer views captured uniformly around the object. Toldo et al. (2009) reported a retrieval algorithm to describe 3D shapes based on articulation-invariant descriptors derived from segmented subparts. More recently, Wang et al. (2010) proposed Intrinsic Spin Images (ISIs) generalizing the traditional spin images (Johnson and Hebert 1999) from 3D space to N-dimensional intrinsic shape space, in which ISIs shape descriptors are computed from MDS embedding representations of original 3D shapes.

Another intuitive solution is to employ topological structures to quantify the similarity between deformable 3D objects. Hilaga et al. (2001) proposed the Topology Matching technique to establish the similarity estimation by comparing their Multiresolutional Reeb Graphs (MRGs), while Sundar et al. (2003) compared 3D objects by applying graph matching techniques to match their skeletons. Recently, Tam and Lau (2007) achieved better retrieval performance against (Hilaga et al. 2001) by using topological and geometric features simultaneously.

Isometry-invariant global geometric information has also been explored for the retrieval of non-rigid 3D shapes. For instance, Jain and Zhang (2007) proposed to apply the eigenvalues of geodesic distance matrix, while Reuter et al. (2005) suggested using the Laplace-Beltrami spectra to generate isometry-invariant shape descriptors. Also, Mahmoudi and Sapiro (2009) designed six such signatures based on the distributions of intrinsic distances including diffusion distance, geodesic distance, a curvature weighted distance, etc.

Above-mentioned three kinds of methods typically have poor discrimination power due to their inaccurate representations for 3D shapes. Thereby, some researchers also tried to address the problem of exact dissimilarity computation between non-rigid models. In Memoli and Sapiro (2005), the authors presented a theoretical framework to directly compare non-rigid 3D shapes based on the Gromov-Hausdorff (GH) distance. Then, Mémoli (2007) approximated the GH distance by solving a mass transportation problem that is basically a quadratic optimization problem with linear constraints. Bronstein et al. (2006) formulated the GH distance as a MDS-like continuous optimization problem, leading to a numerically exact calculation of the GH distance between surfaces. Essentially, matching non-rigid shapes directly is an ideal and complete solution for the calculation of their similarity. However, because of its high computational complexity, direct shape matching is impractical for real searching engines that require instant responses for shape comparisons.

As described in Sect. 1, the utilization of canonical forms is considered to be potentially the best solution for non-rigid 3D shape retrieval. This is because, with the help of canonical forms, we can apply almost any algorithm with respect to the feature extraction, shape matching, and object classification of 3D models in the retrieval of non-rigid 3D shapes. As we know, excellent performance, in term of both accuracy and efficiency, has been achieved for rigid 3D shape retrieval. Consequently, the problem of non-rigid 3D shape retrieval can be well resolved, as soon as it is possible to efficiently construct canonical forms with well-preserved features. The idea of generating canonical forms in 3D domain was initially proposed in Elad and Kimmel (2003), where the authors presented an invariant representation for isometric surfaces using MDS embedding of the surface in a small dimensional Euclidean space in which geodesic distances are approximated by Euclidean ones. They investigated three MDS techniques to construct such 3D canonical forms. Other approaches, like Locally Linear Embedding (LLE) (Mateus et al. 2007), Global Point Signatures (GPS) embedding (based on the Laplace-Beltrami operator) (Rustamov 2007), etc., can also be utilized. To examine the effectiveness of their canonical forms, Elad and Kimmel (2003) extracted a moment-based signature from embedded surfaces and tested it via a simple experiment for object classification, while Lian et al. (2010b) developed a non-rigid 3D shape matching framework using the combination of Least Squares MDS and a visual similarity based method, getting a state-of-the-art retrieval accuracy on the McGill articulated 3D shape benchmark (Siddiqi et al. 2008). However, existing methods, which are typically based on embedding procedures, often obtain greatly distorted canonical forms, and thus could not provide satisfactory performance to distinguish many non-rigid models. This paper addresses the problem by proposing a feature-preserved 3D canonical form. It should be pointed out that, besides shape retrieval, our canonical forms can also be utilized in many other applications including the registration (Huang et al. 2008), classification (Philipp-Foliguet et al. 2011) and recognition (Bronstein et al. 2010) of non-rigid 3D objects.

3 Mathematical Background

In this section, we present a review of the Multidimensional Scaling (MDS) technique that plays an important rule in our feature-preserved canonical form computing method. We first introduce the general idea of MDS algorithms, and then briefly review two MDS methods (i.e., Classical MDS and Least Squares MDS). Finally, advantages and disadvantages of those traditional MDS techniques are discussed.

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 small-dimensional Euclidean space. More specifically, MDS map each feature $Y_i, i = 1, \dots, N$ to its corresponding point $X_i, i = 1, \dots, N$ in a m-dimensional Euclidean space $\mathbf{R}^m$ by minimizing, for example, the following stress function:

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

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 $\mathbf{R}^m$, and w_{ij} is the weighting coefficient for specific applications. Since the canonical forms we want should appear as 3D models, the dimension of the Euclidean space is selected as $m = 3$. Furthermore, in order to make our canonical forms invariant against isometric transformations, we choose the geodesic distance $d_G(Y_i, Y_j)$ as the dissimilarity measure between two features.

Essentially, MDS methods all aim to minimize the difference between $d_F(Y_i, Y_j)$ and $d_E(X_i, X_j)$, which can be expressed using functions like $E_S(X)$. According to the different minimization algorithms applied, existing MDS techniques can be categorized into Classical MDS (Cox and Cox 1994), Least Squares MDS (Borg and Groenen 1997), Fast MDS (Faloutsos and Lin 1995), etc. Here we only briefly review the first two MDS methods that are compared and utilized in this paper. For more details about MDS techniques, we refer the reader to (Cox and Cox 1994; Borg and Groenen 1997; Faloutsos and Lin 1995; Elad and Kimmel 2003; Bronstein et al. 2008).

3.1 Classical MDS

To implement the Classical MDS method. We first need to compute the distance (i.e., dissimilarity) $d_F(Y_i, Y_j)$ between every two features Y_i and Y_j in the feature space, and construct the squared feature distance matrix D_F by

$$D_F = \begin{bmatrix} d_F^2(Y_1, Y_1) & \cdots & d_F^2(Y_1, Y_N) \\ \vdots & \ddots & \vdots \\ d_F^2(Y_N, Y_1) & \cdots & d_F^2(Y_N, Y_N) \end{bmatrix}. \tag{2}$$

D_F is a symmetric matrix since $d_F(Y_i, Y_j) = d_F(Y_j, Y_i)$. Then, we get the inner product matrix (i.e., Gram matrix) G_E in the embedded Euclidean space by

$$G_E = -\frac{1}{2} J D_F J, \tag{3}$$

where

$$J = I - \frac{1}{N}\mathbf{1}\mathbf{1}^T \tag{4}$$

in which I is a $N \times N$ identity matrix and $\mathbf{1}$ is a vector containing N elements that are all 1, namely, $\mathbf{1}_{N\times 1} = [1, 1, \dots, 1]^T$.

After applying singular value decomposition, the matrix G_E can be expressed as

$$G_E = Q\Lambda Q^T, \tag{5}$$

where the diagonal matrix $\Lambda_{N\times N} = diag(\lambda_1, \lambda_2, \ldots, \lambda_k, 0, \ldots, 0)$ and the eigenvalues of G_E are ordered so that $\lambda_1 \geq \lambda_2 \geq \cdots \lambda_k \geq 0$.

Let the dimension of the embedded Euclidean space be m and $m \leq k$, the inner product matrix G_E can be approximated by

$$G_E = UU^T = \begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_N \end{bmatrix} \begin{bmatrix} X_1^T & X_2^T & \cdots & X_N^T \end{bmatrix}. \tag{6}$$

Then we have

$$U = \begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_N \end{bmatrix} = Q_{N\times m}\Lambda_{m\times m}^{\frac{1}{2}}, \tag{7}$$

where $\Lambda_{m\times m} = diag(\lambda_1, \lambda_2, \ldots, \lambda_m)$, $Q_{N\times m}$ denotes the matrix consisting of eigenvectors corresponding to the eigenvalues in $\Lambda_{m\times m}$, and $X_i = [x_1, x_2, \ldots, x_m]$, $i = 1, 2, \ldots, N$ are the coordinates of points generated by using the classical MDS embedding algorithm to map corresponding features in the feature space to the m-dimensional Euclidean space. Basically, instead of the stress function $E_S(X)$ (Eq. (1)), classical MDS minimizes the following energy function

$$E_{S1}(X) = \left\| Q(\Lambda - \tilde{\Lambda})Q^T \right\|^2, \tag{8}$$

where $\| \bullet \|$ denotes the square root of the sum of the squared matrix elements, and $\tilde{\Lambda}_{N\times N} = diag(\lambda_1, \lambda_2, \ldots, \lambda_m, 0, \ldots, 0)$.

3.2 Least Squares MDS

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) (Borg and Groenen 1997) algorithm to minimize the stress function $E_S(X)$. Here, we briefly review the SMACOF algorithm.

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

$$E_{S2}(X) = \sum_{i=1}^{N}\sum_{j=i+1}^{N} w_{ij}\big(d_F(Y_i, Y_j) - d_E(X_i, X_j)\big)^2 \tag{9}$$

or

$$E_{S2}(X) = \varphi_F^2 + \varphi_E^2(X) - 2\phi(X), \tag{10}$$

where

$$\varphi_F^2 = \sum_{i=1}^{N}\sum_{j=i+1}^{N} w_{ij} d_F^2(Y_i, Y_j), \tag{11}$$

$$\varphi_E^2(X) = \sum_{i=1}^{N}\sum_{j=i+1}^{N} w_{ij} d_E^2(X_i, X_j), \tag{12}$$

$$\phi(X) = \sum_{i=1}^{N}\sum_{j=i+1}^{N} w_{ij} d_F(Y_i, Y_j) d_E(X_i, X_j). \tag{13}$$

Utilizing the Cauchy-Schwartz inequality and several basic algebraic operations, we have

$$\begin{aligned} E_{S2}(X) &\leq \varphi_F^2 + trace\big(X^T \Gamma X\big) - 2trace\big(X^T B(\tilde{X})\tilde{X}\big) \\ &= \sigma(X, \tilde{X}), \end{aligned} \tag{14}$$

where $\tilde{X}$ is the approximation of X, namely, a possible solution to minimize the stress function $E_{S2}(X)$, the elements of the matrix $B(\tilde{X})$ are defined by

$$b_{ij} = \begin{cases} -\frac{w_{ij} d_F(Y_i, Y_j)}{d_E(\tilde{X}_i, \tilde{X}_j)}, & i \neq j \quad \text{and} \quad d_E(\tilde{X}_i, \tilde{X}_j) \neq 0 \\ 0, & i \neq j \quad \text{and} \quad d_E(\tilde{X}_i, \tilde{X}_j) = 0, \end{cases} \tag{15}$$

$$b_{ii} = \sum_{j=1, j\neq i}^{N} b_{ij}, \tag{16}$$

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

$$\Gamma = \sum_{i=1}^{N}\sum_{j=i+1}^{N} w_{ij} E_{ij}, \tag{17}$$

$$E_{ij} = (e_i - e_j)(e_i - e_j)^T, \tag{18}$$

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

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

$$\frac{\partial \sigma(X, \tilde{X})}{\partial X} = 2\Gamma X - 2B(\tilde{X})\tilde{X} = 0, \tag{19}$$

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

$$X^{(k)} = \Gamma^{+} B(\tilde{X})\tilde{X}, \tag{20}$$

where Γ^{+} is the Moore-Penrose inverse of Γ. By setting all weights w_{ij} to 1, Eq. (20) can be rewritten as

$$X^{(k)} = \frac{1}{N} B\big(X^{(k-1)}\big)X^{(k-1)}. \tag{21}$$

In practice, given a threshold ε, calculating Eq. (21) iteratively until $E_{S2}(X^{(k)}) - E_{S2}(X^{(k-1)}) < \varepsilon$, we obtain the final solution $X^{(k)}$ for the nonlinear minimization problem of the stress function $E_S(X)$.

3.3 Discussion

Classical MDS is widely considered as an efficient way to solve MDS problems. It can be calculated in $O(n^2)$, where n denotes the number of features that are embedded into the Euclidean space $\mathbf{R}^m$. However, compared to original models, serious distortions (see Fig. 1(b)) are generated on corresponding 3D canonical forms when using the classical MDS approach. Mainly due to the fact that smaller value (see Elad and Kimmel 2003) of the stress function (1) can be obtained by applying Least Squares MDS, the qualities of their canonical forms can be improved (see Fig. 1(c)) compared to classical MDS. But the complexity of the method increases to $O(n^2 \times N_{Iter})$, where N_{Iter} denotes the number of iterations. Generally speaking, all these MDS methods try to use the distance between every two points in the Euclidean space $\mathbf{R}^m$ to approximate the dissimilarity between corresponding pair of features in the feature space. Moreover, they treat all pairs of features in the same manner, no matter how big or small the dissimilarities between them are. However, for example, to build a unique 3D canonical form for a 3D mesh while preserving its important local details, Euclidean distances between pairs of original points in local regions should be equal (or approximately equal) to the corresponding Euclidean distances on the canonical form, and every pair of points with a large geodesic distance (e.g., above a given threshold) should be transformed so that the geodesic distance between them on the original mesh can be approximated by the corresponding Euclidean distance between the transformed points on the canonical form. To achieve this goal, one possible solution is to set the weights w_{ij} in the stress function adaptively according to the values of $d_F(Y_i, Y_j)$. But that introduces a new challenging problem about how to dynamically and precisely relate w_{ij} to $d_F(Y_i, Y_j)$. Another possible solution is to consider MDS embedded surfaces as references and then naturally deform the original meshes against them. This paper utilizes the second approach to construct feature-preserved canonical forms for non-rigid 3D watertight meshes.

4 The Feature-Preserved 3D Canonical Form

In this section, we first briefly describe the framework of our method, and then elaborate on the details of each step in the corresponding subsection.

The strategy of our method is to construct canonical forms by naturally deforming original models to the poses that correspond well with their MDS embedding results. As depicted in Fig. 2, given a 3D mesh, its feature-preserved canonical form can be obtained by using our algorithm which consists of the following three steps:

1. *Initialization*: Reduce the number of vertices on the original surface, and then calculate the initial canonical form by applying Least Squares MDS embedding on the simplified mesh.
2. *Segmentation*: Decompose the original mesh into a set of near-rigid subparts, and then map the segmentation result to the simplified mesh and its embedded surface.

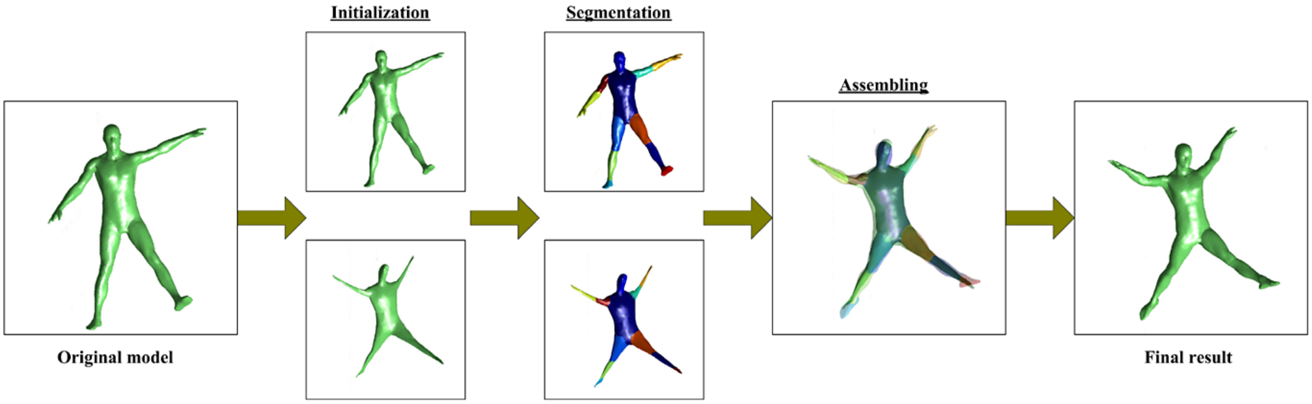


Fig. 2 Overview of our method that consists of the following three steps: Initialization, Segmentation, and Assembling

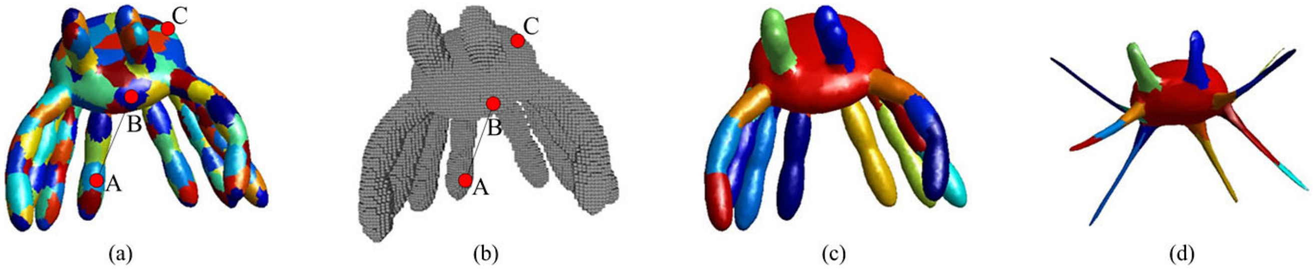


Fig. 3 A demonstration of our mesh segmentation procedure that employs the initial over-segmentation (**a**) and the voxelized model (**b**) to generate final segmentation results for the original model (**c**) and its initial canonical form (**d**)

3. *Assembling*: Register subparts of the original mesh to corresponding components on the embedded surface, and then smooth the segmentation boundaries between subparts on the final canonical form.

4.1 Initialization

Due to the fact that geodesic distances on a surface are insensitive to isometric transformations, a bending invariant representation can be calculated by applying MDS to map the geometric structure of the original surface to a new 3D Euclidean space, in which geodesic distances are approximated by Euclidean ones. This idea was originally presented in Elad and Kimmel (2003), where three MDS techniques were also compared. To calculate the initial canonical form, here, we choose the Least Squares MDS technique implemented by the SMACOF algorithm (Borg and Groenen 1997), which results in the least distortions among those three methods (Elad and Kimmel 2003). A Matlab source code for the Least Squares MDS method, which is publicly available on the web site of the book (Bronstein et al. 2008), is adopted in this paper.

Since the calculation of geodesic distances and the implementation of Least Squares MDS are both time-consuming, the 3D mesh should be simplified to some extent before further processing. A Matlab function called “*reducepatch()*” is utilized in our implementation to reduce the number of vertices on the mesh to about 2000. To sum up, the aim of this step is to generate the simplified mesh and the initial canonical form for a given 3D model.

4.2 Segmentation

Automatic segmentation of 3D models is a fundamental problem in computer graphics (Chen et al. 2009). Until recently, a large amount of methods have been developed to segment 3D meshes into a set of disjoint pieces, which should be either meaningful subparts or ones that satisfy some specifically desired criteria. For more details, we refer the reader to the paper (Chen et al. 2009), in which a survey and comparisons of several mesh segmentation algorithms are presented.

For our purpose of constructing feature-preserved canonical form, we need to decompose the deformable 3D mesh into a number of near-rigid subparts that are convex or approximately convex. To address this problem, we first segment the surface into a large number (e.g., 200) of patches (see Fig. 3(a)). And then, we merge two conjunctive patches in case the convexity of the combined one is above a given threshold (e.g., 0.85). Iterating this procedure until stable, small patches are clustered into several large pieces (see Fig. 3(c)), which can be considered as near-rigid components of the mesh.

A random walk based mesh segmentation method proposed by Lai et al. (2008) is utilized in our algorithm to generate the initial segmentation for the mesh. Here, the original implementation developed by the authors (Lai et al. 2008) with default parameters is directly used without modification. As described in Zunic and Rosin (2004), the convexity of a shape S is defined to be the probability that for randomly chosen points E and F from S all points from the line segment $[EF]$ also belong to S. However, directly determining whether all points on the line $[EF]$ are inside the model is unacceptably time-consuming. Therefore, we voxelize the object to accelerate the computation of convexity. More specifically, given two points on the surface, the line between them is first voxelized in the same manner as the voxelization of the mesh. And then, by judging whether all voxels on the line segment coincide with some voxels of the mesh, we can determine whether the line segment belong to the mesh or not. Examples are given in Fig. 3, where all points on the line segment $[BC]$ belong to the mesh while $[AB]$ is not completely inside the object. It should be pointed out that the voxelization of the closed mesh is accomplished using a reliable source code download from the web site (Morris 2006) and the long-axis resolution is experimentally selected as 150.

Owing to the discretization error introduced by voxelization, there may exist a few small pieces that can not be merged properly. Thereby, we also set a threshold regarding area to merge those small pieces into large subparts that share boundaries with them. Figure 3(c) shows the final segmentation result, which, as we can see, corresponds reason-

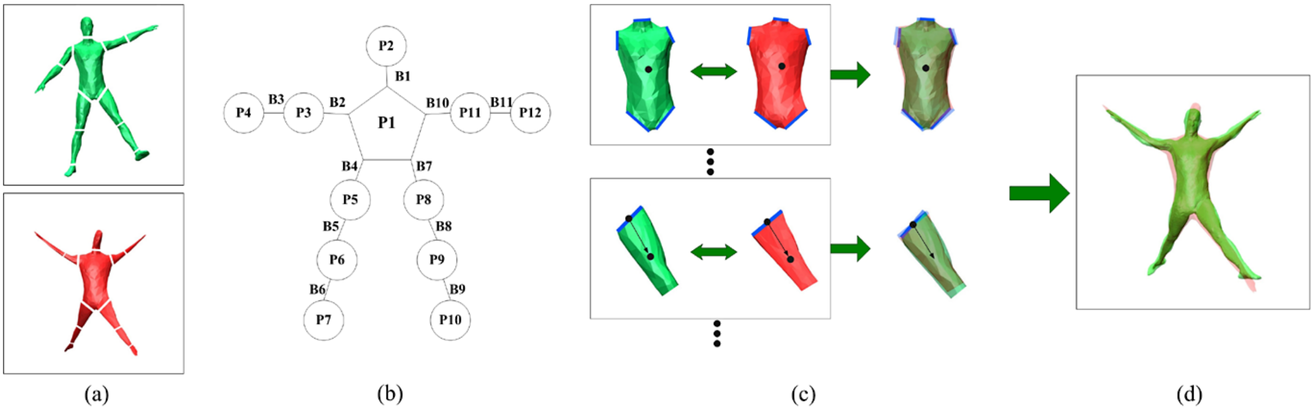


Fig. 4 Overview of the assembling procedure

ably well with intuition. At the end of this step, the segmentations of the simplified mesh and the initial canonical form (see Fig. 3(d)) are also obtained by simply classifying triangle faces into their nearest neighbors (i.e., considering the distance between centers of two triangles) on the original model. As demonstrated by experimental results, our mesh segmentation method provides an adequate solution for our purpose of constructing feature-preserved canonical forms. However, we note that it is neither the only, nor necessarily the best, mesh segmentation method for this task. Better segmentation algorithms can be utilized to further improve the smoothness of surfaces for the models' feature-preserved canonical forms.

4.3 Assembling

The last and the most important step in our method is to assemble segmented subparts to create a new 3D model under the pose that matches well with the initial canonical form (i.e., the embedded surface calculated in the first step).

An overview of the assembling procedure is illustrated in Fig. 4. After mesh segmentation, a part tree $\{P_i, B_j | 1 \leq i \leq N_P, 1 \leq j \leq N_B\}$ is built, where P_i denotes the subpart i, the boundary j is denoted as B_j, and all subparts are connected through boundaries. We define the subpart with maximum number of boundaries as *core part* from which our assembling starts. Given core parts of the simplified mesh M_s and the embedded surface M_e, which are denoted as CP_s and CP_e, respectively, we first translate the mass center of CP_s to that of CP_e, and then rotate CP_s around its center to find the optimal alignment with CP_e. An intuitive explanation for the registration of subparts is given in Fig. 4(c), where the simplified mesh is colored in green, the embedded surface in red, and blue for boundaries. More specifically, after translation, let the vertices of boundaries on the core part CP_s and CP_e be denoted by Vs_i and Ve_i, $i = 1, 2, \dots, N_{vcp}$, respectively. By successively rotating CP_s around x, y, z coordinate axes with angles (α, β, γ), we get new coordinates, represented as $Vs_i' = R(\alpha, \beta, \gamma)Vs_i, i = 1, 2, \dots, N_{vcp}$, for vertices on its boundaries. To achieve the optimal registration between two core parts, we apply the Gauss-Newton algorithm to solve the following minimization problem:

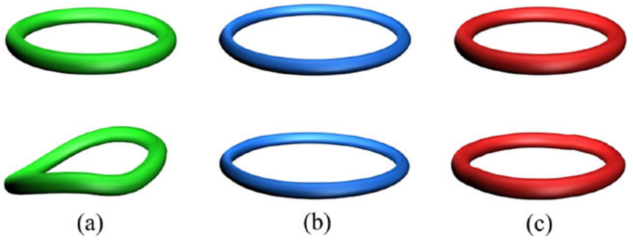


Fig. 5 A perfect torus model and a bent torus model (**a**), together with their 3D canonical forms obtained using Least Squares MDS (**b**) and our method (**c**), respectively

$$\min_{\alpha,\beta,\gamma} \sum_{i=1}^{N_{vcp}} \left\| R(\alpha, \beta, \gamma)Vs_i - Ve_i \right\|^2, \tag{22}$$

where the norm $\| \bullet \|$ is the square root of the sum of the squared matrix elements. In other words, the goal of this non-linear least squares problem is to minimize the sum of squared distances between vertices on the boundaries of the simplified mesh and their corresponding vertices on the embedded surface. It should be pointed out that, as shown in Fig. 5, our method is also able to build feature-preserved canonical forms for objects with circular part trees. Moreover, different choices of core parts due to varied segmentation results have very little impact on the results of our method. For example, any segmented subpart of the torus models shown in Fig. 5(a) can be selected as the core part for our assembling procedure, and the difference between final canonical forms we obtained is negligible.

Once the alignment of two core parts is finished, we no longer need the simplified mesh. The core part of the original object is translated and rotated in the same manner as

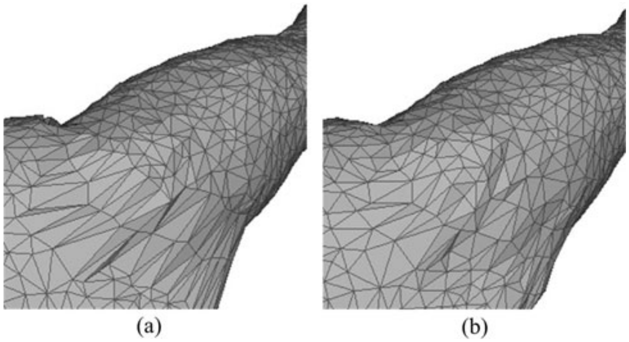


Fig. 6 An example of our results without (**a**) and with (**b**) the simple boundary smoothing method

its simplified version, other subparts (named *child parts*) are then attached to the registered core part one by one according to the structure of the part tree. Generally speaking, the assembling of a child part is comprised of two steps: *coarse alignment* and *precise alignment*. In the stage of coarse alignment, the child part of the original model is first translated to move the center of vertices on one of its boundary (named *fixed boundary*), which also belongs to other registered subparts, to the center of corresponding boundary on the embedded surface. Next, we rotate the original child part around the fixed boundary's center against the embedded child part, so that the direction of the line, which starts from the center of the fixed boundary and ends at the original child part's center, coincides with that of the embedded child part. For a more intuitive demonstration, we refer the reader to the second row in Fig. 4(c). During precise alignment, we aim to find the optimal pose for the original child part such that the boundary between two subparts could be as smooth as possible. Specifically, after coarse alignment, given the child part P_c which is to be assembled and its parent subpart P_p whose location has already been fixed, we denote the vertices on the fixed boundary of P_c and these vertices on P_p as Vc_i and Vp_i, $i = 1, 2, \ldots, N_{vfb}$, respectively. In order to obtain the optimal assembling, the child part P_c is first rotated by angle δ around the line casting from the center of the fixed boundary on P_c to the child part's mass center, and then translated by $T = [t_x, t_y, t_z]^T$. We represent the transformed coordinates of Vc_i by

$$Vc_i' = R_L(\delta) Vc_i + T, \tag{23}$$

where $R_L(\delta)$ stands for the rotation mentioned above, while T means translating vertices along x, y, and z axes by t_x, t_y, and t_z, respectively. Then, the precise alignment can be formulated as the following minimization problem:

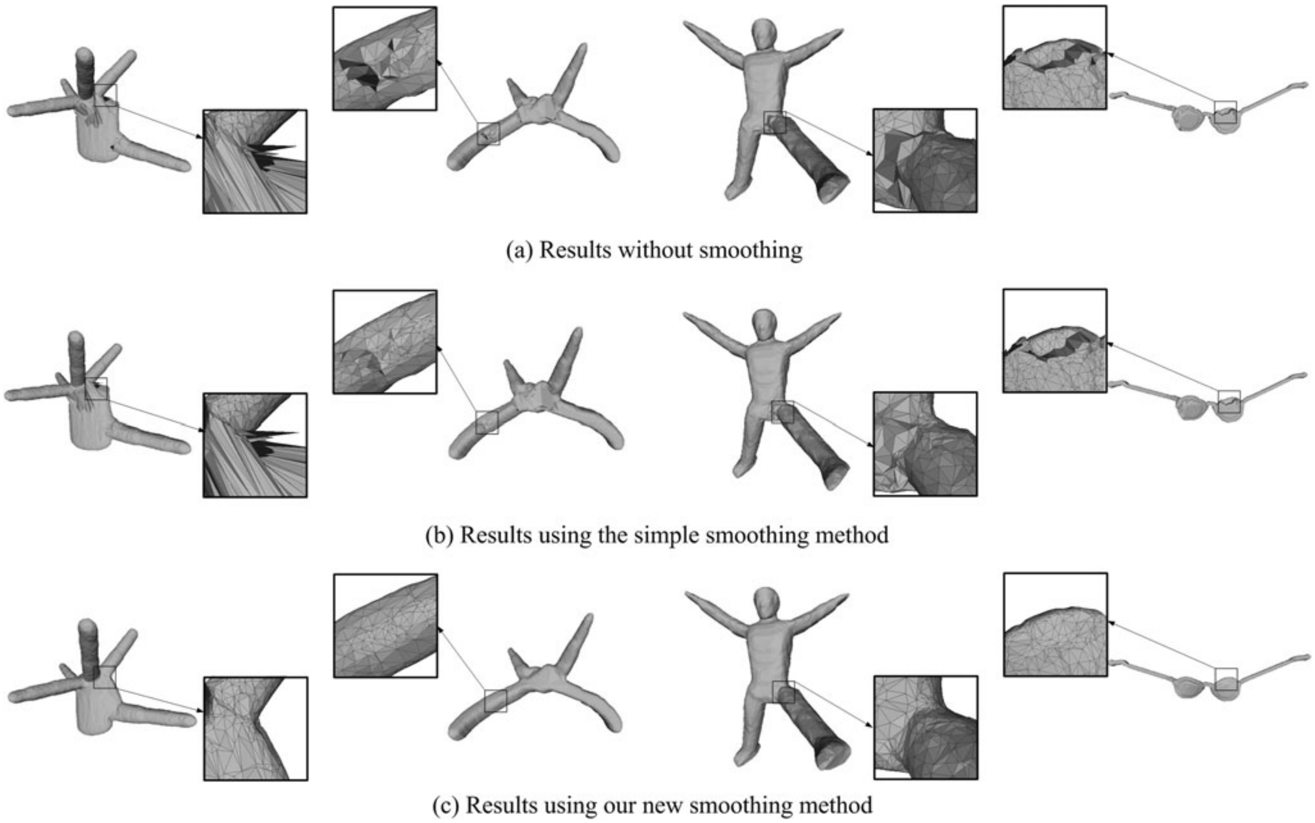
$$\min_{t_x, t_y, t_z, \delta} \sum_{i=1}^{N_{vfb}} \left\| R_L(\delta) Vc_i + T - Vp_i \right\|^2. \tag{24}$$

Calculating the mean values of vertices Vc_i' and Vp_i, $i = 1, 2, \ldots, N_{vfb}$, we obtain a new boundary between the subpart P_c and P_p. However, simply doing so yields a uneven region (see Figs. 6(a) and 7(a)). To smooth the boundary, our previous paper (Lian and Godil 2011) proposed a simple method as follows. First, we uniformly divide the vertices around the boundary into N_g groups based on the distance value between each vertex and its nearest neighbor on the boundary. More precisely, group 1 contains vertices that are closest to the boundary, group 2 are the second closest group around the boundary, and so for every group k, $k = 1, 2, \ldots, N_g$. Next, we move vertices on the classified groups continuously towards their nearest neighbors on the boundary. To be specific, given a vertex Vg_* in group k and its closest vertex (Vc_{i*}' or Vp_{i*}) on the boundary, the new coordinate for Vg_* is defined as

$$Vg_*' = \begin{cases} Vg_* + \frac{Vc_{i*}' - Vp_{i*}}{2^k}, & \text{if } Vg_* \in P_c \\ Vg_* - \frac{Vc_{i*}' - Vp_{i*}}{2^k}, & \text{if } Vg_* \in P_p. \end{cases} \tag{25}$$

We show an example of our results using the above simple boundary smoothing method in Fig. 6(b), which clearly corresponds better with intuition than the unprocessed result (Fig. 6(a)). However, as we can see from Fig. 7(b), for unsmoothed canonical forms that contains large gaps on the boundaries between subparts, the simple smoothing algorithm does not work well. Although in some cases (e.g., the plier and human models shown in Fig. 7) results can be improved reasonably by applying the simple smoothing method (Fig. 7(b)) compared to unprocessed ones (Fig. 7(a)), usually they are still unsatisfactory for visualization, shape matching, and many other applications. Inspired by the ideas of *As-Rigid-As-Possible Shape Interpolation* (Alexa et al. 2000) and *Deformation Transfer* (Sumner and Popovic 2004), we design a new and effective approach to solve the boundary smoothing problem for all kinds of objects. As shown in Fig. 7(c), since the deformation from a mesh to its 3D canonical form is guaranteed to be as-rigid-as-possible by using our new smoothing algorithm, final results we obtain are quite natural and smooth.

The boundary smoothing problem can be stated as follows: Given a 3D mesh M which consists of N_f triangles $\{T_1, T_2, \ldots, T_{N_f}\}$ and N_v vertices $\{V_1, V_2, \ldots, V_{N_v}\}$, after applying the above-mentioned assembling procedure, its unsmoothed canonical form M' can be obtained that contains N_f triangles $\{T_1', T_2', \ldots, T_{N_f}'\}$ and $3 \cdot N_f$ vertices $\{V_1'(1), V_2'(1), V_3'(1), V_1'(2), V_2'(2), V_3'(2), \ldots, V_1'(N_f), V_2'(N_f), V_3'(N_f)\}$. With M and M', we could like to build the final canonical form $\tilde{M}$ which is composed of N_f triangles $\{\tilde{T}_1, \tilde{T}_2, \ldots, \tilde{T}_{N_f}\}$ and N_v vertices $\{\tilde{V}_1, \tilde{V}_2, \ldots, \tilde{V}_{N_v}\}$. Namely, we want to compute the parameters of the unknown affine transformation for each pair of triangles $\{T_i, \tilde{T}_i\}$ based on the known affine mapping of the triangle pair $\{T_i, T_i'\}$.



(a) Results without smoothing

(b) Results using the simple smoothing method

(c) Results using our new smoothing method

Fig. 7 A comparison of our feature-preserved canonical forms generated without smoothing (**a**), and those created using the simple smoothing method (**b**) and our new smoothing algorithm (**c**), respectively

Let the vertices of the original triangle T_i and the target triangle $\tilde{T}_i$ be $\{V_{i_1}, V_{i_2}, V_{i_3}\}$ and $\{\tilde{V}_{i_1}, \tilde{V}_{i_2}, \tilde{V}_{i_3}\}$, respectively. While, the vertices of T_i' are denoted as $\{V_1'(i), V_2'(i), V_3'(i)\}$. As described in Sumner and Popovic (2004), the three vertices of a triangle before and after deformation do not fully determine the affine transformation due to the fact that they can not provide information about how the space perpendicular to the triangle deforms. Thereby, a new vertex is added for each triangle in the direction perpendicular to it. Specifically, we compute the new vertex for the triangle T_i as

$$V_{N_v+i} = V_{i_1} + \frac{(V_{i_2} - V_{i_1}) \times (V_{i_3} - V_{i_1})}{\sqrt{|(V_{i_2} - V_{i_1}) \times (V_{i_3} - V_{i_1})|}} \qquad (26)$$

and employ an analogous calculation to add $\tilde{V}_{N_v+i}$ for the triangle $\tilde{T}_i$. Also, we define the fourth vertex on the triangle T_i' by

$$V_4'(i) = V_1'(i) + \frac{(V_2'(i) - V_1'(i)) \times (V_3'(i) - V_1'(i))}{\sqrt{|(V_2'(i) - V_1'(i)) \times (V_3'(i) - V_1'(i))|}}. \qquad (27)$$

An affine transformation that deforms T_i into T_i' can be formulated as

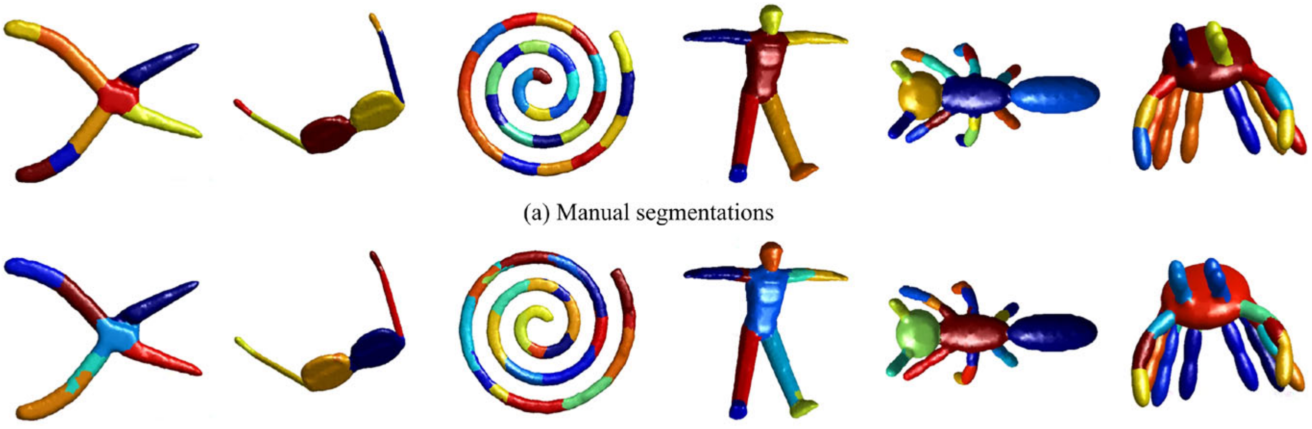
$$\begin{aligned} A_i V_{p(j)} + L_i &= \begin{bmatrix} a_1(i) & a_2(i) & a_3(i) \\ a_4(i) & a_5(i) & a_6(i) \\ a_7(i) & a_8(i) & a_9(i) \end{bmatrix} \begin{bmatrix} x_{p(j)} \\ y_{p(j)} \\ z_{p(j)} \end{bmatrix} \\ &\quad + \begin{bmatrix} l_x(i) \\ l_y(i) \\ l_z(i) \end{bmatrix} \\ &= V_j'(i), \end{aligned} \qquad (28)$$

where $V_j'(i) = [x_j'(i), y_j'(i), z_j'(i)]^T$ and $[p(j)]_{1\times 4} = [i_1, i_2, i_3, N_v + i]$, $j \in \{1, 2, 3, 4\}$. Similarly, for the affine mapping from T_i to $\tilde{T}_i$, we have

$$\begin{aligned} \tilde{A}_i V_f + \tilde{L}_i &= \begin{bmatrix} \tilde{a}_1(i) & \tilde{a}_2(i) & \tilde{a}_3(i) \\ \tilde{a}_4(i) & \tilde{a}_5(i) & \tilde{a}_6(i) \\ \tilde{a}_7(i) & \tilde{a}_8(i) & \tilde{a}_9(i) \end{bmatrix} \begin{bmatrix} x_f \\ y_f \\ z_f \end{bmatrix} + \begin{bmatrix} \tilde{l}_x(i) \\ \tilde{l}_y(i) \\ \tilde{l}_z(i) \end{bmatrix} \\ &= \tilde{V}_f, \end{aligned} \qquad (29)$$

where $\tilde{V}_f = [\tilde{x}_f, \tilde{y}_f, \tilde{z}_f]^T$, $f \in \{i_1, i_2, i_3, N_v + i\}$.

Subtracting the first equation from the other three for Eq. (28) and Eq. (29), respectively, we get $A_i D_i = D_i'$ and



(a) Manual segmentations

(b) Automatic segmentations

Fig. 8 Examples of mesh segmentations generated manually by a human being (**a**) and automatically by our method (b), respectively

$\tilde{A}_i D_i = \tilde{D}_i$ where

$$D_i = [V_{i_2} - V_{i_1}, V_{i_3} - V_{i_1}, V_{N_v+i} - V_{i_1}], \tag{30}$$

$$D'_i = \left[V'_2(i) - V'_1(i), V'_3(i) - V'_1(i), V'_4(i) - V'_1(i)\right], \tag{31}$$

$$\tilde{D}_i = [\tilde{V}_{i_2} - \tilde{V}_{i_1}, \tilde{V}_{i_3} - \tilde{V}_{i_1}, \tilde{V}_{N_v+i} - \tilde{V}_{i_1}]. \tag{32}$$

Trivially, we have $A_i = D'_i D_i^{-1}$ and $\tilde{A}_i = \tilde{D}_i D_i^{-1}$. Here, A_i is the known matrix for the affine transformation that deforms T_i to T'_i, while elements of the unknown matrix $\tilde{A}_i$ are linear combinations of the coordinates of vertices on the final canonical form to be constructed. Locations of those unknown vertices can be determined by minimizing the quadratic error between the original transformation matrices A_i and the final ones $\tilde{A}_i$. We write the minimization problem as

$$\min_{\tilde{V}_1,\ldots,\tilde{V}_{N_v},\tilde{V}_{N_v+1},\ldots,\tilde{V}_{N_v+N_f}} \sum_{i=1}^{N_f} \|A_i - \tilde{A}_i\|^2. \tag{33}$$

In order to have a unique solution for Eq. (33), we treat a vertex, say $\tilde{V}_1$, as a constant rather than as a free variable. By setting $\tilde{U}^T = [\tilde{x}_2, \tilde{y}_2, \tilde{z}_2, \ldots, \tilde{x}_{N_v+N_f}, \tilde{y}_{N_v+N_f}, \tilde{z}_{N_v+N_f}]$ and $C^T = [\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_{N_f}]$ where $\mathbf{c}_i = [a_1(i), a_2(i), \ldots, a_9(i)]$, the minimization problem (Eq. (33)) can be rewritten as

$$\min_{\tilde{x}_2,\tilde{y}_2,\tilde{z}_2,\ldots,\tilde{x}_{N_v+N_f},\tilde{y}_{N_v+N_f},\tilde{z}_{N_v+N_f}} \|C - H\tilde{U}\|^2, \tag{34}$$

in which H is a sparse matrix of size $9N_f \times (3N_v + 3N_f - 3)$ that relates $\tilde{U}$ to C, and the elements of H can be easily derived according to the following expression:

$$\tilde{A}_i = \tilde{D}_i D_i^{-1} = \begin{bmatrix} \tilde{x}_{i_2} - \tilde{x}_{i_1} & \tilde{x}_{i_3} - \tilde{x}_{i_1} & \tilde{x}_{N_v+i} - \tilde{x}_{i_1} \\ \tilde{y}_{i_2} - \tilde{y}_{i_1} & \tilde{y}_{i_3} - \tilde{y}_{i_1} & \tilde{y}_{N_v+i} - \tilde{y}_{i_1} \\ \tilde{z}_{i_2} - \tilde{z}_{i_1} & \tilde{z}_{i_3} - \tilde{z}_{i_1} & \tilde{z}_{N_v+i} - \tilde{z}_{i_1} \end{bmatrix} D_i^{-1}. \tag{35}$$

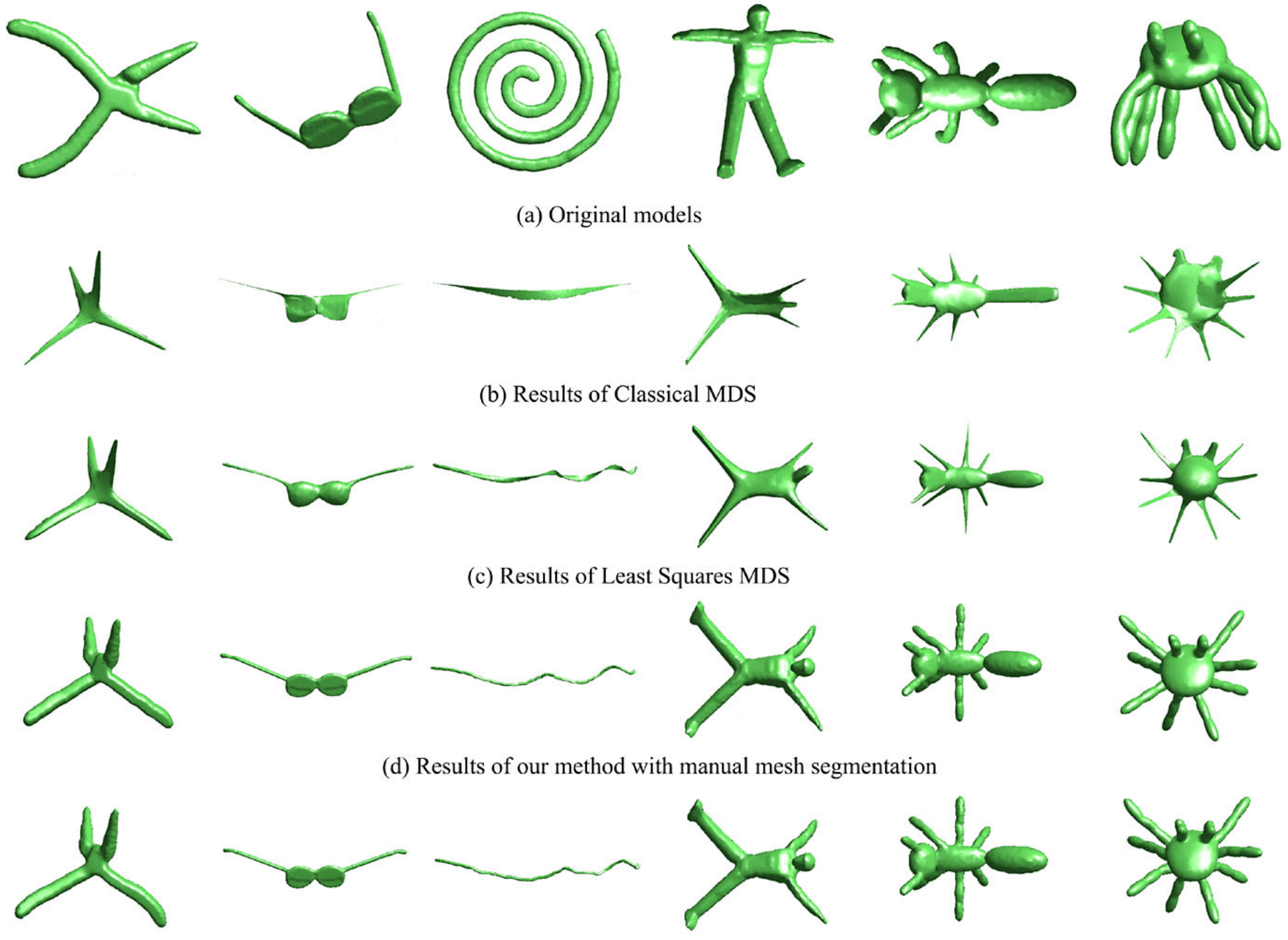
We solve the minimization problem (Eq. (34)) by setting the gradient of the objective function over the free variable $\tilde{U}$ to zero:

$$H^T H\tilde{U} = H^T C. \tag{36}$$

In practice, H is typically a large-scale matrix (e.g., for a mesh with 5000 vertices and 10000 triangles, the size of H is 90000 × 44997), and thus, the computational cost for direct calculation of $(H^T H)^{-1}$ is often unacceptable. Therefore, we apply a sparse QR solver (Davis 2011) to efficiently solve Eq. (36). Final results of several models using our new boundary smoothing method are demonstrated in Fig. 7(c). Compared to the previous results shown in Figs. 7(a) and (b), complete surfaces of the final canonical forms are smooth and natural, even if there are uneven segmentation boundaries and huge gaps between connected subparts. That makes our feature-preserved canonical form computing method considerably robust against the quality of mesh segmentation result. Moreover, experimental results presented below also demonstrate that the new boundary smoothing algorithm can help to improve performance for methods using our feature-preserved canonical forms in the application of non-rigid 3D shape retrieval.

5 Experimental Results

In order to validate the effectiveness of our method, feature-preserved 3D canonical forms were calculated for all 255



(a) Original models

(b) Results of Classical MDS

(c) Results of Least Squares MDS

(d) Results of our method with manual mesh segmentation

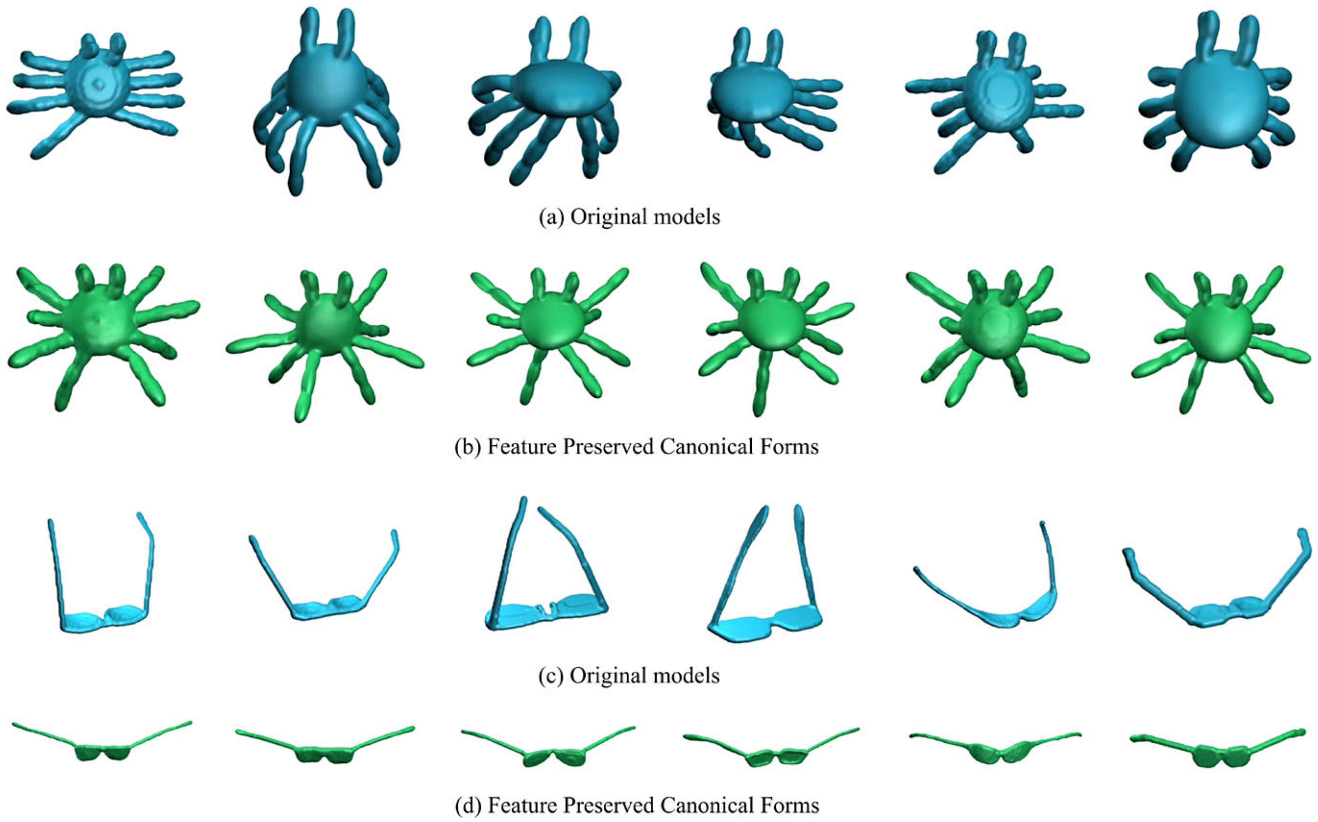
(e) Results of our method with automatic mesh segmentation

Fig. 9 Examples of 3D models (**a**) and their canonical forms (**b**) (**c**) (**d**) (**e**) constructed using four methods

models in the McGill articulated 3D shape database (Siddiqi et al. 2008), which is classified into 10 categories. We implemented the algorithm in Matlab R2009 and carried out experiments run under windows XP on a personal computer with a 3.19 GHz Intel Xeon CPU, 12.0 GB DDR2 memory, and a 512 MB NVIDIA Quadro Fx580 graphics card. The average time to create our canonical form for a model, which contains 10000 triangle faces on average, is about 243 seconds, in which, approximately, we spent 120 seconds in computing the embedded surface with 4000 triangle faces, took 63 seconds to segment the mesh, and finished the assembling step in 60 seconds.

In Fig. 8, our mesh segmentation results are compared with human-generated segmentations, which are created by a researcher using the Interactive Segmentation tool developed by Chen et al. (2009). When doing manual segmentation, the researcher is required to segment each 3D model into a number of near-rigid subparts. We find that the automatic segmentation approach generally produces results as good as the manual one, but some segmentation boundaries of our approach are still not so even due to the utilization of patch clustering. As suggested in Lai et al. (2008), similar boundary smoothing techniques could be applied to further improve segmentation results. Thanks to the utilization of our new smoothing method, we can still obtain satisfactory canonical forms even if the segmentation results contain some minor errors (see Fig. 7).

Figure 9 shows several examples of our feature-preserved canonical forms using automatic segmentation (Fig. 9(e)) as well as manual segmentation (Fig. 9(d)), along with their original models (Fig. 9(a)) and embedded surfaces calculated by classical MDS (Fig. 9(b)) and Least Squares MDS (Fig. 9(c)), respectively. As we can see, results obtained using MDS embedding techniques are seriously distorted, while our new canonical forms not only provide isometry-invariant representations for non-rigid 3D meshes, but also preserve important features that appear on the original models. We also observe that, our feature-preserved canonical forms constructed using the method with automatic segmentation are almost identical with the results generated by the


(a) Original models

(b) Feature Preserved Canonical Forms

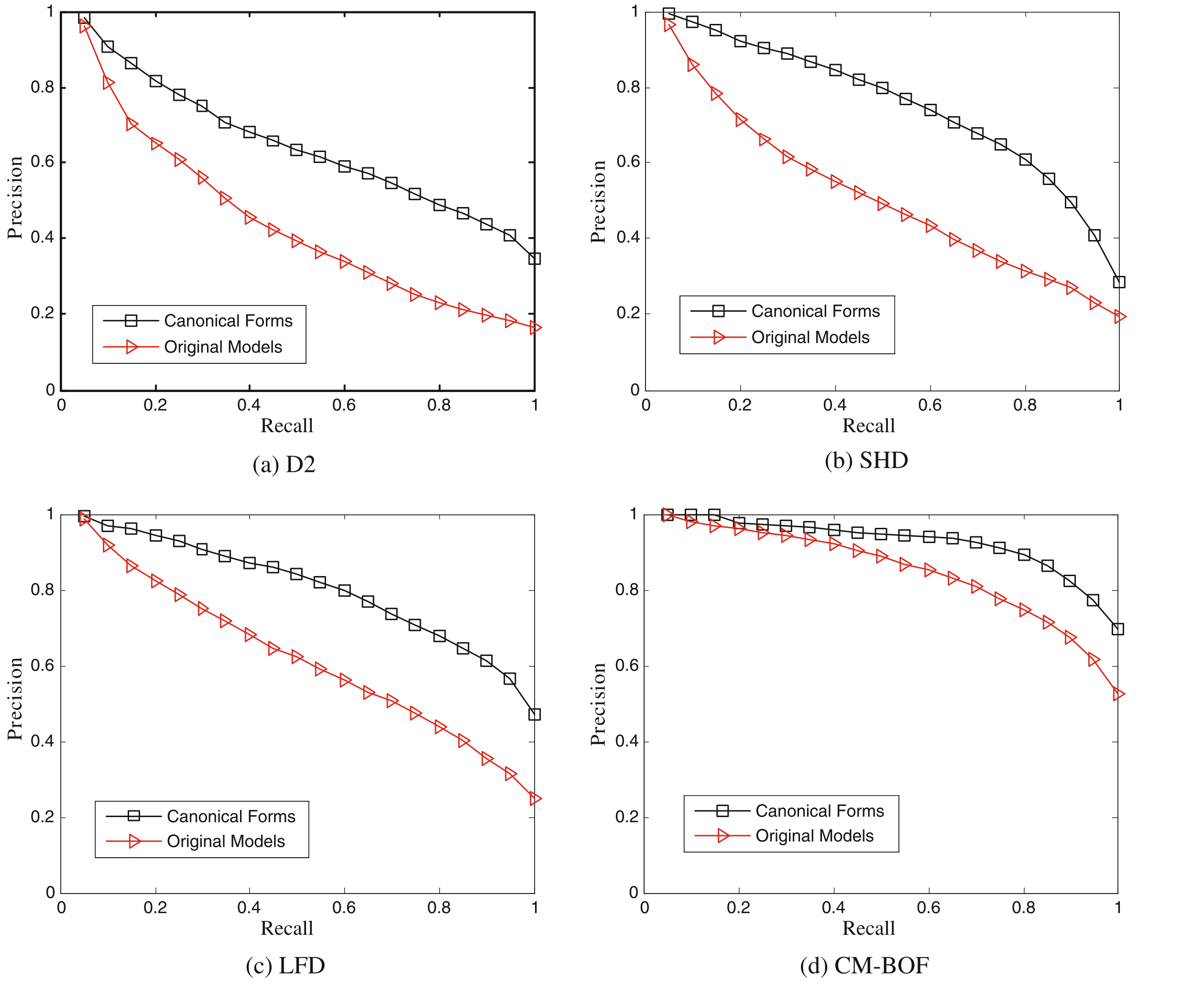
(c) Original models

(d) Feature Preserved Canonical Forms

Fig. 10 Examples of two kinds of 3D models (**a**) (**c**) and their feature-preserved canonical forms (**b**) (**d**)

approach with manual segmentation. That again validates the robustness of our canonical form computing method against segmentation errors.

We then demonstrate the application of our feature-preserved canonical form in non-rigid 3D shape retrieval. Experiments were carried out on the McGill articulated 3D shape benchmark (Siddiqi et al. 2008). As we can see from Fig. 10, models in the same class may appear in quite different poses but can still have very similar canonical forms. Thereby, after the calculation of canonical forms, all feature extraction methods, which are even specifically designed for rigid 3D shapes, can be employed to extract isometry-invariant shape descriptors from non-rigid objects. Figure 11 illustrates the precision-recall plots of several well-known rigid shape retrieval methods (i.e., D2 Osada et al. 2002, SHD Kazhdan et al. 2003, LFD Chen et al. 2003, and CM-BOF Lian et al. 2010a) evaluated on two databases that consist of original models on the McGill database and their feature-preserved canonical forms, respectively. We observe that, with the help of our feature-preserved canonical forms, significant improvements can be achieved for the performance of these methods in the application of non-rigid 3D shape retrieval. In order to take advantage of the preserved local details on our canonical form as well as its isometry-invariant global structure, here we adopted two visual similarity based approaches called CM-BOF (Lian et al. 2010a) and Hybrid (Lian et al. 2010a), respectively, with default settings. To be specific, CM-BOF utilizes salient local features to describe views captured around a 3D model, while the Hybrid method, which represents 2D views by using both local and global features, is basically a combination of the CM-BOF and GSMD (Lian et al. 2010c) methods. For convenience, the Feature-Preserved Canonical Form (FPCF) obtained by our method with Automatic mesh segmentation is denoted as AFPCF. Therefore, AFPCF-Hybrid and AFPCF-CM-BOF stand for the retrieval methods using the Hybrid and CM-BOF approaches, respectively, with AFPCF canonical forms. While LSMDS-CM-BOF denotes the approach using CM-BOF with Least Squares MDS embedding and CMDS-CM-BOF for the Classical MDS method. Precision-recall plots as well as four quantitative measures (NN, 1-Tier, 2-Tier, DCG) (Shilane et al. 2004) were calculated to compare the searching accuracy of the proposed approaches (i.e., AFPCF-Hybrid and AFPCF-CM-BOF) with the following 8 non-rigid 3D shape retrieval methods: LSMDS-CM-BOF (Lian et al. 2010b), CMDS-CM-BOF, BF-SIFT (Ohbuchi et al. 2008), Intrinsic Spin Images (ISI8) (Wang et al. 2010), Heat Kernel Signatures (HKS) (Ovsjanikov et al. 2009), Spin Images (SI) (Johnson and Hebert 1999), the shape distribution of Geodesic



(a) D2

(b) SHD

(c) LFD

(d) CM-BOF

Fig. 11 Precision-recall plots of four rigid 3D shape retrieval methods evaluated on two versions of the McGill articulated 3D shape database that consist of original models and their feature-preserved canonical forms, respectively

distance (G2) (Mahmoudi and Sapiro 2009), and Laplace-Beltrami Spectrum (LBS) (Reuter et al. 2005). As we can see from the results demonstrated in Fig. 12 and Table 1, methods using our feature-preserved canonical form generally provide more accurate searching results than those using other canonical forms. Moreover, the AFPCF-Hybrid algorithm that employs the proposed canonical form markedly outperform the state of the art of non-rigid 3D shape retrieval. Yet, in despite of advantages on all other measures, the 2-Tier value of AFPCF-CM-BOF is 0.9 % less than LSMDS-CM-BOF. We speculate that the task of searching models in the McGill database is relatively easy and the searching accuracy of the original CM-BOF method is already quite high, thus taking account of more local details in shape matching can only slightly further improve the retrieval accuracy for the CM-BOF method tested on this small non-rigid 3D shape database. We show some examples of queries and their corresponding top 7 retrieved objects from the McGill database using our AFPCF-CM-BOF algorithm in Fig. 13. As we can see, the retrieved 3D models in the top 7 positions of the rank lists all belong to the same categories of their corresponding queries, which again demonstrates the effectiveness of the proposed feature-preserved

Table 1 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
AFPCF-Hybrid	**99.6 %**	**87.5 %**	**96.5 %**	**97.8 %**
AFPCF-CM-BOF	**99.6 %**	**86.6 %**	**95.3 %**	**97.5 %**
LSMDS-CM-BOF	99.2 %	84.8 %	96.2 %	97.4 %
CMDS-CM-BOF	96.1 %	74.2 %	88.6 %	93.9 %
BF-SIFT	97.3 %	74.6 %	87.0 %	93.7 %
ISI8	95.3 %	64.2 %	79.9 %	90.0 %

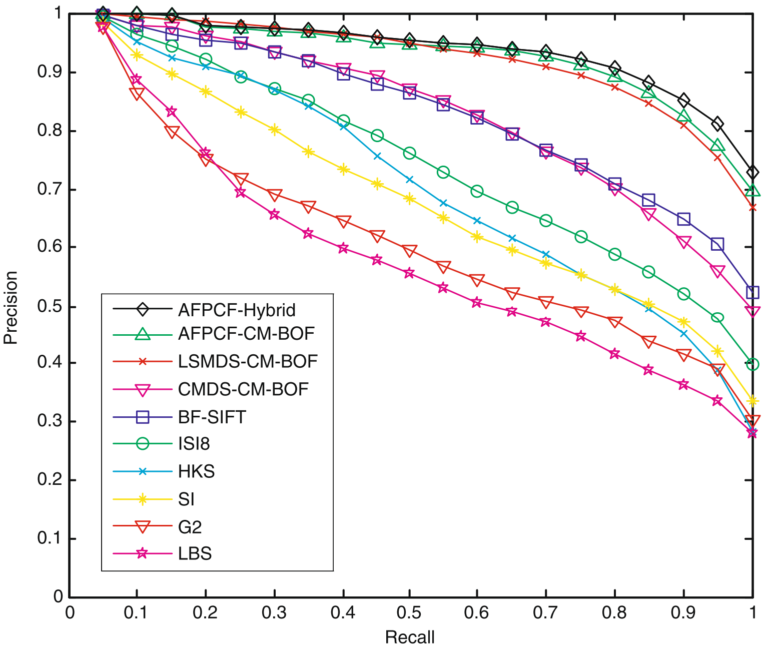


Fig. 12 Precision-recall plots of the proposed methods (i.e., AFPCF-Hybrid and AFPCF-CM-BOF) and other eight approaches evaluated on the McGill articulated 3D shape database

canonical form in the application of non-rigid 3D shape retrieval.

From Fig. 13, we also observe that different types of objects in the McGill articulated 3D shape database have quite different topologies, and thus detailed local features do not play an important role in the dissimilarity computation between non-rigid 3D models in this database. That is the major reason why methods using our feature-preserved canonical forms can only slightly outperform the approaches that utilize Least Squares MDS embedding. To demonstrate more clearly the advantages of our feature-preserved canonical forms against other existing methods, we built the Peking University Non-rigid 3D Shape Benchmark (PKUNSB), on which additional experiments were carried out. The PKUNSB database contains 90 articulated 3D watertight meshes, which are equally classified into six categories. Examples of these six kinds of objects are shown in Fig. 14, from which we can see that they have similar skeletons but differ in details on surfaces. In order to precisely distinguish each articulated model from other types of objects in the PKUNSB database, shape descriptors that we extract from the object not only should be invariant against isometric transformations but should also be able to depict important local features on the surface. In our non-rigid 3D shape retrieval experiments conducted using PKUNSB, the CM-BOF method was evaluated on five databases that consist of original models in the PKUNSB database and their four types of 3D canonical forms, respectively. For the sake of convenience, we denote the feature-preserved canonical form with the simple boundary smoothing approach as S-AFPCF, and we use the above-mentioned notations including AFPCF, LSMDS, and CMDS for other three kinds of canonical forms, respectively. From Fig. 15 and Table 2 where experimental results are shown, we observe that methods using feature-preserved canonical forms (i.e., AFPCF and S-AFPCF) significantly outperform those using other two types of canonical forms (i.e., LSMDS and CMDS) and original models (i.e., Original). Furthermore, the CM-BOF method using our canonical forms with the improved boundary smoothing algorithm (i.e., AFPCF) performs better than the one with the simple smoothing approach (i.e., S-AFPCF). This is due to the fact that the proposed feature-preserved canonical form (i.e., AFPCF) computing method not only preserves important local features on the original model, but also eliminates distortions and errors on the boundaries between segmented subparts.

Table 2 Comparing retrieval results of the CM-BOF method evaluated on five versions of the PKUNSB database that consist of original models (i.e., Original) and their four kinds of 3D canonical forms (i.e., AFPCF, S-AFPCF, LSMDS, and CMDS), respectively

	NN	1-Tier	2-Tier	DCG
AFPCF	**98.9 %**	**89.0 %**	**99.5 %**	**97.3 %**
S-AFPCF	97.8 %	88.0 %	99.5 %	97.1 %
LSMDS	97.8 %	81.7 %	97.1 %	95.8 %
CMDS	97.8 %	73.4 %	92.2 %	93.8 %
Original	92.2 %	64.9 %	83.2 %	89.2 %

6 Discussion and Conclusion

In this paper, we introduced a novel method to construct feature-preserved canonical forms for non-rigid 3D meshes.

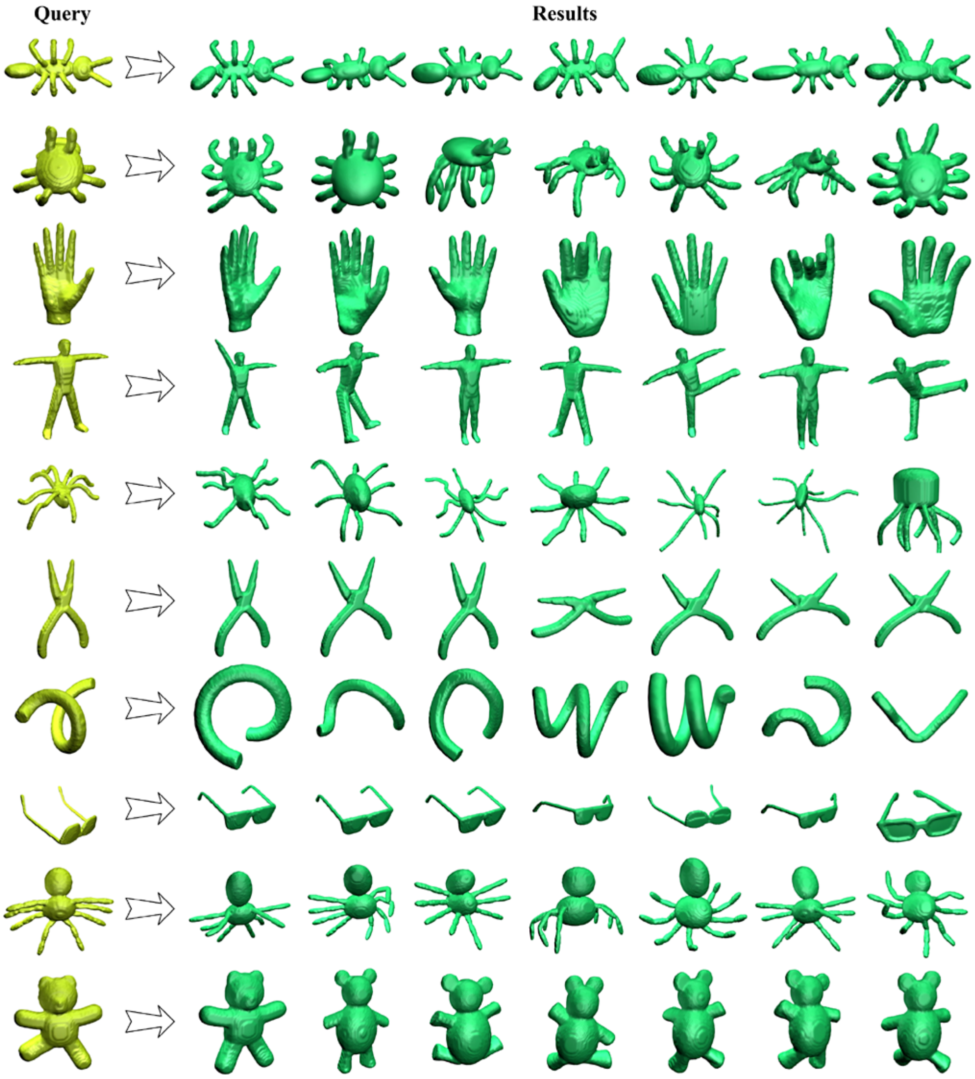


Fig. 13 Examples of queries (first column) selected from the McGill articulated 3D shape database and their corresponding top 7 retrieved models using the AFPCF-CM-BOF method. The retrieved models are ranked from left to right based on the increasing order of dissimilarity

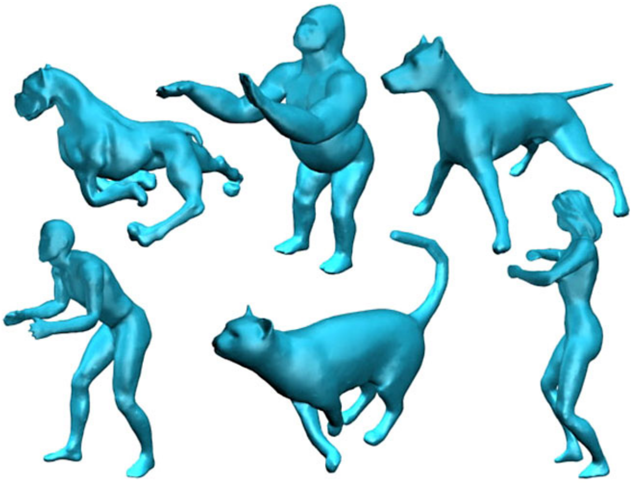


Fig. 14 Examples of models in the PKUNSB database, which contains 90 watertight meshes that are generated by articulating six original 3D objects

Experimental results not only validated the advantages of our method against other existing canonical forms, but also demonstrated the effectiveness of the algorithm for non-rigid 3D shape retrieval, in the presence of better performance compared to the state of the art.

Nevertheless, there are also limitations in our method. First, the method can only be applied on 3D watertight meshes and it is sensitive against topological errors. Second, although the method is quite robust against mesh segmentation results, it still relates closely to the segmentation procedure such that large segmentation errors on some bad-quality surfaces may lead to distortions on our feature-preserved canonical forms. Third, the computational cost of the method we implemented for our experiments is too expensive to establish instant responses which are required by real searching engines. That can be solved to some extent by optimizing our Matlab code and developing more efficient algorithms to accomplish MDS embedding.

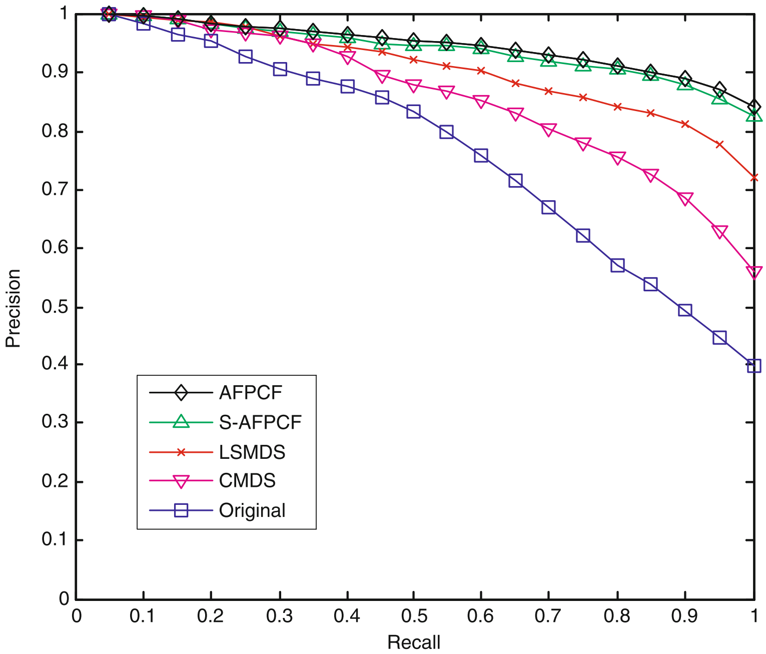


Fig. 15 Precision-recall plots of the CM-BOF method evaluated on five versions of the PKUNSB database that consist of original models (i.e., Original) and their four kinds of 3D canonical forms (i.e., AFPCF, S-AFPCF, LSMDS, and CMDS), respectively

Four directions we would like to consider in the future are listed as follows: (1) Develop more effective and efficient algorithms to segment 3D meshes into near-rigid parts; (2) Employ other mesh manipulation methods to naturally deform 3D models; (3) Carry out experiments for the proposed canonical form on other benchmarks that contain larger numbers of non-rigid 3D objects; (4) Utilize our feature-preserved canonical form in other applications for non-rigid 3D shapes including registration (Chui and Rangarajan 2003), correspondences finding (Wuhrer et al. 2010), object classification (Elad and Kimmel 2003), and so on.

Acknowledgements This work has been supported by China Postdoctoral Science Foundation (Grant No.: 2012M510274), the SIMA program and the Shape Metrology IMS. We would like to thank the anonymous reviewers for their constructive comments, and Xu-Lei Wang for providing his results that have been compared in this paper.

References

Alexa, M., Cohen-Or, D., & Levin, D. (2000). As-rigid-as-possible shape interpolation. In *Proc. SIGGRAPH* (pp. 157–164).

Borg, I., & Groenen, P. (1997). *Modern multidimensional scaling—theory and applications*. Berlin: Springer.

Bronstein, A. M., Bronstein, M. M., & Kimmel, R. (2006). Efficient computation of isometry-invariant distances between surfaces. *SIAM Journal on Scientific Computing*, *28*(5), 1812–1836.

Bronstein, A. M., Bronstein, M. M., & Kimmel, R. (2008). *Numerical geometry of non-rigid shapes*. Berlin: Springer.

Bronstein, A. M., Bronstein, M. M., Kimmel, R., Mahmoudi, M., & Sapiro, G. (2010). A Gromov-Hausdorff framework with diffusion geometry for topologically-robust non-rigid shape matching. *International Journal of Computer Vision*, *89*(2–3), 266–286.

Chen, D. Y., Tian, X. P., Shen, Y. T., & Ouhyoung, M. (2003). On visual similarity based 3D model retrieval. In *Proc. Eurographics 2003* (pp. 223–232).

Chen, X., Golovinskiy, A., & Funkhouser, T. (2009). A benchmark for 3D mesh segmentation. *ACM Transactions on Graphics*, *28*(3). doi:10.1145/1531326.1531379.

Chui, H., & Rangarajan, A. (2003). A new point matching algorithm for non-rigid registration. *Computer Vision and Image Understanding*, *89*(2–3), 114–141.

Cox, M. A., & Cox, T. F. (1994). *Multidimensional scaling*. London/New York: Chapman and Hall.

Davis, T. A. (2011). Algorithm 915, SuiteSparseQR: multifrontal multithreaded rank-revealing sparse QR factorization. *ACM Transactions on Mathematical Software*, *38*(1). doi:10.1145/2049662.2049670.

Elad, A., & Kimmel, R. (2003). On bending invariant signatures for surface. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *25*(10), 1285–1295.

Faloutsos, C., & Lin, K. D. (1995). A fast algorithm for indexing, datamining and visualisation of traditional and multimedia datasets. In *Proc. ACM SIGMOD* (pp. 163–174).

Funkhouser, T., & Shilane, P. (2006). Partial matching of 3D shapes with priority-driven search. In *Proc. symposium on geometry processing (SGP'06)* (pp. 131–142).

Gal, R., Shamir, A., & Cohen-Or, D. (2007). Pose-oblivious shape signature. *IEEE Transactions on Visualization and Computer Graphics*, *13*(2), 261–271.

Hilaga, M., Shinagawa, Y., Kohmura, T., & Kunii, T. L. (2001). Topology matching for fully automatic similarity estimation of 3D shapes. In *Proc. SIGGRAPH* (pp. 203–212).

Huang, Q., Adams, B., Wicke, M., & Guibas, L. J. (2008). Non-rigid registration under isometric deformations. *Computer Graphics Forum*, *27*(5), 1449–1457.

Jain, V., & Zhang, H. (2007). A spectral approach to shape-based retrieval of articulated 3D models. *Computer Aided Design*, *39*(5), 398–407.

Johnson, A. E., & Hebert, M. (1999). Using spin images for efficient object recognition in cluttered 3D scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *21*(5), 433–449.

Kazhdan, M., Funkhouser, T., & Rusinkiewicz, S. (2003). Rotation invariant spherical harmonic representation of 3D shape descriptors. In *Proc. symposium on geometry processing (SGP'03)* (pp. 156–164).

Lai, Y., Hu, S., Martin, R. R., & Rosin, P. L. (2008). Fast mesh segmentation using random walks. In *Proc. SPM'08* (pp. 183–191).

Lian, Z., & Godil, A. (2011). A feature-preserved canonical form for 3D meshes. In *Proc. international conference on 3D imaging, modeling, processing, visualization and transmission (3DIMPVT'11)* (pp. 116–123).

Lian, Z., Godil, A., & Sun, X. (2010a). Visual similarity based 3D shape retrieval using bag-of-features. In *Proc. shape modeling international (SMI'10)* (pp. 25–36).

Lian, Z., Godil, A., Sun, X., & Zhang, H. (2010b). Non-rigid 3D shape retrieval using multidimensional scaling and bag-of-features. In *Proc. international conference on image processing (ICIP 2010)* (pp. 3181–3184).

Lian, Z., Rosin, P. L., & Sun, X. (2010c). Rectilinearity of 3D meshes. *International Journal of Computer Vision*, *89*(2–3), 130–151.

Liu, Y., Zha, H., & Qin, H. (2006). Shape topics: a compact representation and new algorithms for 3D partial shape retrieval. In *Proc. IEEE conference on computer vision and pattern recognition (CVPR'06)* (pp. 2025–2032).

Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, *60*(2), 91–110.

Mahmoudi, M., & Sapiro, G. (2009). Three-dimensional point cloud recognition via distributions of geometric distances. *Graphical Models*, *71*(1), 22–31.

Mateus, D., Cuzzolin, F., Horaud, R., & Boyer, E. (2007). Articulated shape matching using locally linear embedding and orthogonal alignment. In *Proc. international conference on computer vision (ICCV'07)* (pp. 1–8).

Memoli, F. (2007). On the use of Gromov-Hausdorff distances for shape comparison. In *Proc. SPG'07* (pp. 81–90).

Memoli, F., & Sapiro, G. (2005). A theoretical and computational framework for isometry invariant recognition of point cloud data. *Foundations of Computational Mathematics*, *5*(3), 313–347.

Morris, D. (2006). Voxelizer: floodfilling and distance map generation for 3D surfaces. http://techhouseorg/~dmorris/projects/voxelizer/.

Ohbuchi, R., Osada, K., Furuya, T., & Banno, T. (2008). Salient local visual features for shape-based 3D model retrieval. In *Proc. shape modeling international (SMI'08)* (pp. 93–102).

Osada, R., Funkhouser, T., Chazelle, B., & Dobkin, D. (2002). Shape distributions. *ACM Transactions on Graphics*, *21*(4), 807–832.

Ovsjanikov, M., Bronstein, A. M., Bronstein, M. M., & Guibas, L. (2009). Shape google: a computer vision approach to isometry invariant shape retrieval. In *Proc. NORDIA'09* (pp. 320–327).

Philipp-Foliguet, S., Jordan, M., Najman, L., & Cousty, J. (2011). Artwork 3D model database indexing and classification. *Pattern Recognition*, *44*(3), 588–597.

Reuter, M., Wolter, F. E., & Peinecke, N. (2005). Laplace-spectra as fingerprints for shape matching. In *Proc. SPM'05* (pp. 101–106).

Rustamov, R. M. (2007). Laplace-Beltrami eigenfunctions for deformation invariant shape representation. In *Proc. symposium on geometry processing (SGP'07)* (pp. 225–233).

Shilane, P., Min, P., Kazhdan, M., & Funkhouser, T. (2004). The Princeton shape benchmark. In *Proc. shape modeling international (SMI'04)* (pp. 167–178).

Siddiqi, K., Zhang, J., Macrini, D., Shokoufandeh, A., Bouix, S., & Dickinson, S. (2008). Retrieving articulated 3D models using medial surfaces. *Machine Vision and Applications*, *19*(4), 261–275.

Sumner, R. W., & Popovic, J. (2004). Deformation transfer for triangle meshes. In *Proc. SIGGRAPH* (pp. 399–405).

Sun, J., Ovsjanikov, M., & Guibas, L. (2009). A concise and provably informative multi-scale signature based on heat diffusion. In *Proc. symposium on geometry processing (SGP'09)* (pp. 1383–1392).

Sundar, H., Silver, D., Gagvani, N., & Dickinson, S. (2003). Skeleton based shape matching and retrieval. In *Proc. shape modeling international (SMI'03)* (pp. 130–139).

Tam, G., & Lau, R. (2007). Deformable model retrieval based on topological and geometric signatures. *IEEE Transactions on Visualization and Computer Graphics*, *13*(3), 470–482.

Tangelder, J. W., & Veltkamp, R. C. (2008). A survey of content based 3D shape retrieval methods. *Multimedia Tools and Applications*, *39*(3), 441–471.

Toldo, R., Castellani, U., & Fusiello, A. (2009). Visual vocabulary signature for 3D object retrieval and partial matching. In *Proc. Eurographics workshop on 3D object retrieval (3DOR'09)* (pp. 21–28).

Wang, X., Liu, Y., & Zha, H. (2010). Intrinsic spin images: a subspace decomposition approach to understanding 3D deformable shapes. In *Proc. international symposium on 3D data processing, visualization and transmission (3DPVT'10)* (pp. 17–20).

Wuhrer, S., Azouz, Z. B., & Shu, C. (2010). Posture invariant surface description and feature extraction. In *Proc. IEEE conference on computer vision and pattern recognition (CVPR'10)* (pp. 374–381).

Zunic, J., & Rosin, P. L. (2004). A new convexity measure for polygons. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *26*(7), 923–934.