Eurographics Workshop on 3D Object Retrieval (2017)
I. Pratikakis, F. Dupont, and M. Ovsjanikov (Editors)

3D Mesh Unfolding via Semidefinite Programming

Juncheng Liu , Zhouhui Lian† and Jianguo Xiao

Institute of Computer Science and Technology, Peking University, PR China

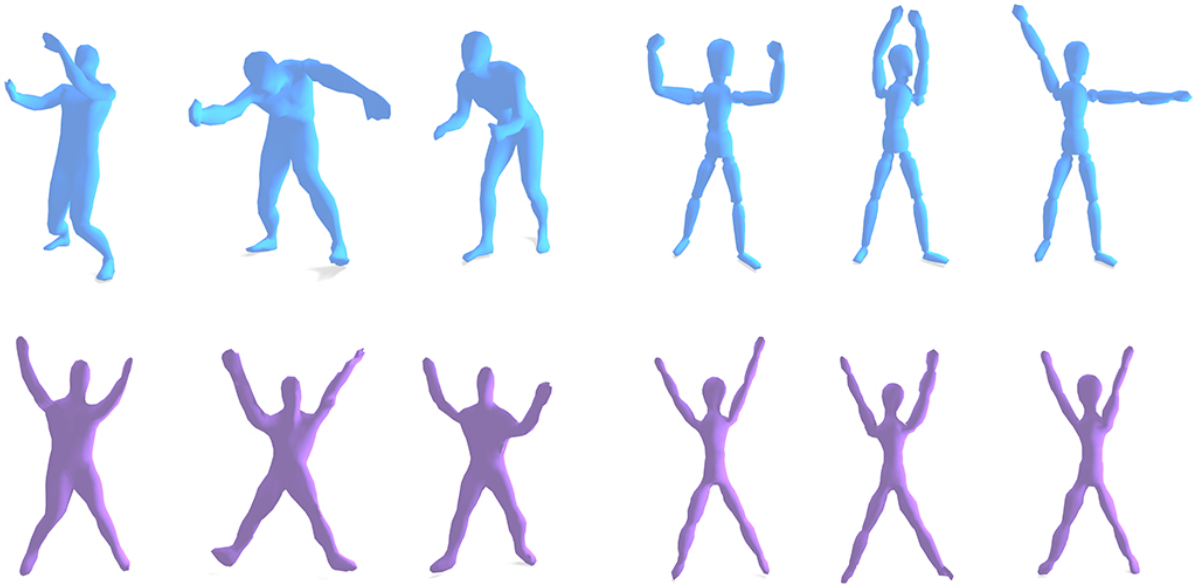

Figure 1: *Mesh unfolding. Top: isometric 3D models with different poses. Bottom: our method unfolds the meshes into isometry-invariant canonical poses which facilitate subsequent tasks including shape retrieval, segmentation, matching, etc.*

Abstract

Mesh unfolding is a powerful pre-processing tool for many tasks such as non-rigid shape matching and retrieval. Shapes with articulated parts may exist large variants in pose, which brings difficulties to those tasks. With mesh unfolding, shapes in different poses can be transformed into similar canonical forms, which facilitates the subsequent applications. In this paper, we propose an automatic mesh unfolding algorithm based on semidefinite programming. The basic idea is to maximize the total variance of the vertex set for a given 3D mesh, while preserving the details by minimizing locally linear reconstruction errors. By optimizing a specifically-designed objective function, vertices tend to move against each other as far as possible, which leads to the unfolding operation. Compared to other Multi-Dimensional Scaling (MDS) based unfolding approaches, our method preserves significantly more details and requires no geodesic distance calculation. We demonstrate the advantages of our algorithm by performing 3D shape matching and retrieval in two publicly available datasets. Experimental results validate the effectiveness of our method both in visual judgment and quantitative comparison.

1. Introduction

With the rapid growth of large 3D shape repositories, efficient ways to organize and explore them become extremely important. Content-based shape retrieval [TV10] has drawn much attention since decades ago and there exist large amounts of work. It pro-

† Corresponding author. E-mail: lianzhouhui@pku.edu.cn

DOI: 10.2312/3dor.20171059

vides an intuitive way of searching similar shapes for queries provided by users. However, for non-rigid 3D shapes, many traditional approaches could fail due to the large variants in pose. Non-rigid shapes are generally composed of articulated parts which are able to transform rigidly. To conduct non-rigid shape retrieval, there are typically three types of methods as follows: 1) locality-based pose-invariant features [SAS07, OOFB08]; 2) global isometry-invariant properties [BB13]; 3) canonical forms. Among these methods, the last one is considered to be the most promising solution due to the fact that it can be utilized as a pre-processing step ahead of the rigid-shape retrieval pipeline, which facilitates the application of existing powerful rigid-shape retrieval techniques into non-rigid cases. Despite its importance, techniques for mesh unfolding are far from being exhaustively explored. Existing techniques are mainly based on the Multi-Dimensional Scaling (MDS) pipeline. This kind of methods requires geodesic calculation and bring severe distortions. Geodesic distance is vulnerable to topological noise and calculating it is very time-consuming.

In this paper, we propose an automatic mesh unfolding method by optimizing an objective function using semi-definite programming. Motivated by the Maximum Variance Unfolding (MVU) [WS04] algorithm, we formulate the total vertex variance to the trace of the Gram matrix. Furthermore, we integrate a locally linear preserving term into the objective function, which leads to an enhancement of the detail-preserving ability. In general, the contributions of this paper lie in the following four aspects: 1) A new 3D mesh unfolding method is proposed without calculating geodesic distances; 2) We formulate the unfolding problem as a form of semidefinite programming and deduce the integration of locally linear preserving term; 3) Our method can be applied to non-watertight meshes with holes while most existing approaches can only handle watertight models; 4) The accuracy of shape matching and retrieval can be improved by employing the proposed method. Our method achieves the state-of-the-art performance among approaches that extract GSMD 3D shape descriptors [LRS10] from various kinds of 3D canonical forms on two public datasets, which suggests the strongest stretching ability.

2. Related Work

Mesh unfolding. The most widely-used 3D mesh unfolding method is realized by leveraging the ISOMAP [TSL00] method. ISOMAP first approximates the geodesic distance by calculating the pairwise shortest paths on the given mesh structure. Then classical Multi-Dimensional Scaling (MDS) is carried out to obtain the coordinates of the unfolded mesh. Alternatively, the classical MDS can be substituted by the least-square MDS [BG03] to obtain unfolding results with less distortions. Recently, [Sah15] proposed to take landmarks of MDS as handles to deform the original mesh. Yet the distortions brought by ISOMAP pipeline algorithms remain serious, and the geodesic distances are computationally intensive and vulnerable. Lian *et al.* [LGX13] proposed the first feature-preserving mesh unfolding method which utilizes the ISOMAP canonical form as guidance and deforms the original mesh into the unfolded pose. The feature-preserving method significantly improves the performance in shape retrieval which verifies the importance of mesh unfolding. Yet, their algorithm requires

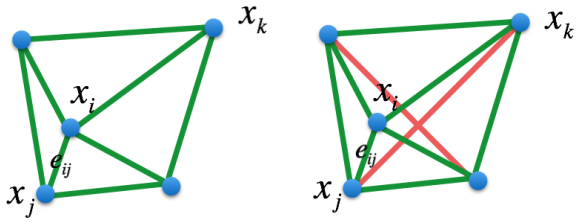

Figure 2: *The neighbor graph defined by mesh topology (left) and the one constructed by MVU algorithm (right). Compared with the mesh topology, MVU constructs a complete graph within the same neighborhood (red edges indicate the augmentations).*

plausible segmentations of meshes. Manual segmentations are time and effort-consuming while auto-segmentation is far from satisfying. The most recent work of mesh unfolding is implemented using finite-elements and deformation [SK16]. Their method is quite different against the ISOMAP pipeline and no calculation of geodesic distance is required. However, their method needs the tetrahedralization of triangle meshes. This disables the method from being used in holey meshes. Furthermore, choosing appropriate parameters for regularization weights in their method is hard since it is difficult to compromise between the unfolding intensity and over-stretching.

Maximum variance unfolding. The Maximum Variance Unfolding (MVU) method [WS04] is first proposed as an approach to discover nonlinear underlying manifold embedded in a high-dimensional space. The basic idea is to maximize the total variance while preserving local structures. The method formulates the total variance within a point set into the trace of the Gram matrix. The length of each edge is preserved by adding constraints to the Gram matrix. Then semidefinite programming techniques are employed to obtain the optimized solution. Compared to ISOMAP, MVU requires no vulnerable and computationally intensive geodesic distances calculation. In MVU, the geodesic distances are approximated by maximizing the overall variance instead of computing the shortest paths, which makes it more robust to topology error and noise.

This paper presents a novel mesh unfolding method that also requires no calculation for geodesic distances. Thereby, our method is robust to mesh noises and holes. It yields comparable visual results as [SK16]. Additionally, since the proposed method operates directly on the given mesh, no additional vertices and triangles are introduced, which facilitates subsequent processing such as feature extraction and registration.

3. Algorithm

In this section, we describe the proposed algorithm in details. We first briefly review the Maximum Variance Unfolding (MVU) method and our modification in order to apply it in mesh unfolding. Then, we improve the original approach to make it more detail-preserving. Specifically, a locally linear preserving term is deduced and integrated into the objective function. With this term, more details on the mesh can be preserved and hence the quality of the unfolded mesh is improved.

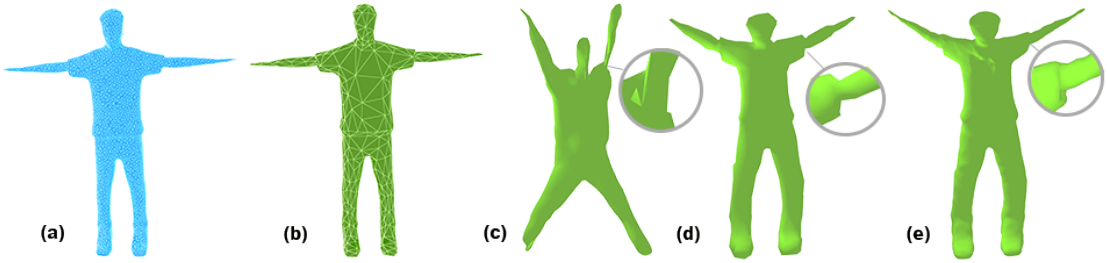

Figure 3: *Demonstration of the effect of locality preserving term. (a) original mesh. (b) the simplified mesh of (a). (c) the unfolded mesh by only maximum variance term (4). (d) the unfolded mesh by locality preserving maximum variance (13) and is restored to original resolution (e).*

3.1. Maximum variance unfolding

The MVU algorithm, which is also called semidefinite embedding. $\mathcal{X} = \{x_1, x_2, \ldots, x_n\}$, where n is the number of points, the MVU method first establishes a neighborhood structure by constructing k nearest neighbor graph. Then it tries to find a set of low-dimensional representation $\mathcal{Y}$, such that $\mathcal{Y}$ keeps the local geometry of $\mathcal{X}$:

$$\mathcal{Y} = \arg\min_{Y} \sum_{i=1}^{n} \sum_{j,k \in N(i)} (\|x_j - x_k\|^2 - \|y_j - y_k\|^2)^2. \tag{1}$$

where $N(i)$ is the index set of neighbors of the point x_i. All pairwise Euclidean distances within the same neighborhood are hence preserved by minimizing the above function. However, solving (1) requires non-convex optimization techniques, which are typically complex and time-consuming. MVU converts the above function to a set of hard constraints and employs semidefinite programming to address this problem.

Let $\mathbf{D}$ denote the Euclidean distance matrix which stores Euclidean distances between all pairs of points in $\mathcal{X}$, i.e. $\mathbf{D}_{ij} = \|x_i - x_j\|$. Similarly, we define the corresponding squared Euclidean distance matrix as $\mathbf{S}$, where $\mathbf{S}_{ij} = \mathbf{D}_{ij}^2$. The overall variance of $\mathcal{X}$ can be represented as $\sum_{ij} \|x_i - x_j\|^2$. With the above definition, it can also be written as $\sum_{ij} D_{ij}^2 = \sum_{ij} S_{ij}$. Assume that $\mathbf{K}$ is the Gram matrix of the given point set $\mathcal{X}$ and $\mathbf{K}$ collects all the pairwise inner product of $\mathcal{X}$, it is easy to verify that the Gram matrix and the Euclidean matrix can be related as follows:

$$\mathbf{D}_{ij}^2 = \mathbf{K}_{ii} + \mathbf{K}_{jj} - 2\mathbf{K}_{ij}. \tag{2}$$

The overall variance then can be formulated using the Gram matrix as:

$$\sum_{ij} \mathbf{D}_{ij}^2 = \|\mathbf{D}\|_{\mathcal{F}}^2 = 2n\, tr(\mathbf{K}) \tag{3}$$

where the $\|\cdot\|_{\mathcal{F}}$ is the frobenius norm. Therefore, the objective function that MVU optimizes is defined as:

$$\begin{aligned} &\text{maximize } tr(\mathbf{K}) \\ &s.t. \sum_{ij} \mathbf{K}_{ij} = 0 \\ &\mathbf{K} \in \mathcal{S}_n^+ \\ &\forall\, i, \mathbf{K}_{k,k} + \mathbf{K}_{j,j} - 2\mathbf{K}_{k,j} = \mathbf{S}_{k,j}, \\ &j,k \in N(i). \end{aligned} \tag{4}$$

Given a mesh to be unfolded, we denote the mesh as a tuple $\mathcal{M} = \{\mathcal{V}, \mathcal{E}\}$, where $\mathcal{V}$ and $\mathcal{E}$ represent the vertex set and edge set, respectively. We regard the vertex set $\mathcal{V}$ as the data point set $\mathcal{X}$ described above. Then the optimization problem introduced in (4) is resolved to produce the unfolded mesh.

Technically, MVU preserves the lengths of edges between all pairs of points within the same neighborhood, which is a complete graph. However, in our experiments, we found that using the classical MVU algorithm fails to produce a plausible mesh, this is due to the fact that constructing the nearest neighbor graph violates the original triangle mesh's topology. As a result, a great amount of edges are unnecessarily preserved. Over stretching and distortions are hence introduced.

To solve this problem, we directly use the structure of the original mesh instead. Namely, we preserve the length of each edge of the triangles of the mesh to be unfolded. Our modified objective function is defined as:

$$\begin{aligned} &\text{maximize } tr(\mathbf{K}) \\ &s.t. \sum_{ij} \mathbf{K}_{ij} = 0 \\ &\mathbf{K} \in \mathcal{S}_n^+ \\ &\forall\ < v_k, v_j > \in \mathcal{E}, \\ &\mathbf{K}_{k,k} + \mathbf{K}_{j,j} - 2\mathbf{K}_{k,j} = \mathbf{S}_{kj}. \end{aligned} \tag{5}$$

In the modified objective function, we preserve the length of each edge for every triangle instead of the constructed edge in the complete graph of original setting (4) as shown in Figure 2. The topology contained in the triangular mesh is far more trustworthy than the neighborhood defined by knn or ε-neighborhood.

After solving the optimization problem (5), the optimized Gram

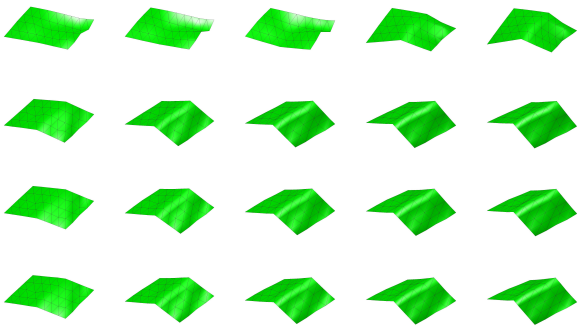

Figure 4: *Unfolding results with different regularization weights of locally linear preserving (left to right:* $\alpha = 1/100, 1/300, 1/500, 1/700$ *, and* $1/900$*) and number of neighbors (top to bottom:* $k = 3, 5, 7$*, and* 9*). The top-left of this matrix approaches the strongest unfolding intensity and weakest detail-preserving, whereas the bottom-right is on the contrary.*

matrix is obtained. Then, We implement the standard Multi-Dimensional Scaling (MDS) or least square MDS to obtain the final coordinates of the unfolded mesh. Note that the topology remains the same during above all processing stages.

3.2. Locally linear structure preserving

In the previous section, the core idea of our algorithm is introduced. However, the unfolded mesh obtained by simply employing (5) is insufficient. As illustrated by Figure 3, some detail features are lost and squashed after optimization (yet the result still looks more natural than those generated by the traditional MDS and LSMDS methods).

To address this issue, we add one term in the objective function to enhance the ability of detail preserving. Inspired by the locally linear embedding (LLE) approach [RS01], we integrate the local linear preserving term into the optimizing function, which in turn preserves the local geometry of the mesh. According to the theorem of barycentric coordinates, each point within a simplex can be represented by the linear combination of the vertices that span the simplex. Specifically, assuming $\mathcal{P} = \{p_1, p_2, \ldots, p_{d+1}\}$ spans the simplex in $\mathcal{R}^d$, an arbitrary point p within it can be represented as $p = \sum_{i=1}^{d+1} w_i p_i$. In matrix format, we have $p = \mathbf{P}w$.

Given a 3D triangle mesh $\mathcal{M} = \{\mathcal{V}, \mathcal{E}\}$, each individual vertex v_i is adjoined to some neighbor vertices (k nearest neighbors). We can therefore define the local geometry by the linear reconstruction coefficients of the neighbor vertices:

$$w_i = \arg\min_a \|v_i - \sum_{j \in N(i)} a_j v_j\|^2, \quad s.t. \quad \sum_{j=1}^{|N(i)|} a_j = 1. \tag{6}$$

We enforce the sum of coefficients to unit. In the above formula, the linear coefficients w_i for each v_i are obtained by minimizing the least square reconstruction error of v_i using the Lagrangian multiplier method. Each w_i is augmented to a length of $|\mathcal{V}|$ by filling zeros to the entries corresponding to non-neighbors. The so-called locally linear embedding error is minimized to obtain an alternative configuration $\mathcal{Y}$ for the vertex set $\mathcal{V}$:

$$\mathcal{Y} = \arg\min_{\mathbf{Y}} \sum_{i=1}^{|\mathcal{V}|} \|y_i - \mathbf{Y}w_i\|^2 \tag{7}$$

under the constraints

$$\mathbf{Y}1 = 0, \quad \mathbf{Y}\mathbf{Y}^T = \mathbf{I}, \tag{8}$$

where $\mathbf{Y} = [y_1, y_2, \ldots, y_n] \in \mathcal{R}^{d \times n}$ is the data matrix of $\mathcal{Y}$. Written in matrix format, the LLE kernel $\mathbf{G}$ is defined as:

$$\mathbf{G} = (\mathbf{I} - \mathbf{W})^T (\mathbf{I} - \mathbf{W}), \tag{9}$$

where the matrix $\mathbf{W}$ stores w_i^T in its rows. With the kernel, we are now able to rewrite (7) as:

$$\mathcal{Y} = \arg\min_{\mathbf{Y}} \ tr(\mathbf{Y}\mathbf{G}\mathbf{Y}^T). \tag{10}$$

Classical LLE performs spectral decomposition of the kernel matrix $\mathbf{G}$ described above. However, we would like to integrate this term in the semidefinite programming framework proposed in Equation (5). To achieve this, we reformulate (10) using the commutative law of trace:

$$\mathcal{Y} = \arg\min_{\mathbf{Y}} \ tr(\mathbf{Y}^T\mathbf{Y}\mathbf{G}) \tag{11}$$

$$= tr(\mathbf{K}\mathbf{G}). \tag{12}$$

where the semidefinite matrix $\mathbf{K}$ is the same as the one defined in the previous section. Minimizing the locally linear reconstruction error reinforces the detail preserving ability of the proposed method. Integrating this term with the above variance maximizing term (5), we finalize the objective function as:

$$\begin{aligned} \underset{\mathbf{K}}{minimize} \ & tr(\mathbf{K}(\mathbf{G} - \alpha\mathbf{I})) \\ s.t. \ & \sum_{ij} \mathbf{K}_{ij} = 0 \\ & \mathbf{K} \in \mathcal{S}_n^+ \\ & \forall \ < v_k, v_j > \in \mathcal{E}, \\ & \mathbf{K}_{k,k} + \mathbf{K}_{j,j} - 2\mathbf{K}_{k,j} = \mathbf{S}_{kj}. \end{aligned} \tag{13}$$

Where α is a weighting parameter balancing the stretching intensity and the property of locality preserving. An intuitive way of understanding how it works is illustrated by Figure 4. Smaller α favors more detail-preserving and weakens the unfolding intensity, while larger α is right on the contrary. In our implementations, we always set α to be $1/100$.

Another parameter influences the results is k, the number of neighbors for reconstruction coefficients calculation. It collaborates with α on the scale of detail-preserving. We found that the influence of choosing different k is not vital as illustrated in Figure 13. In our implementations, we set k to 30 for all models.

4. Experiments

4.1. Datasets and evaluation methods

We implemented the proposed algorithm in Matlab 2014a. To solve the optimization problem (13) we used CVX, a package for specifying and solving convex problems [GB14, GB08]. The running

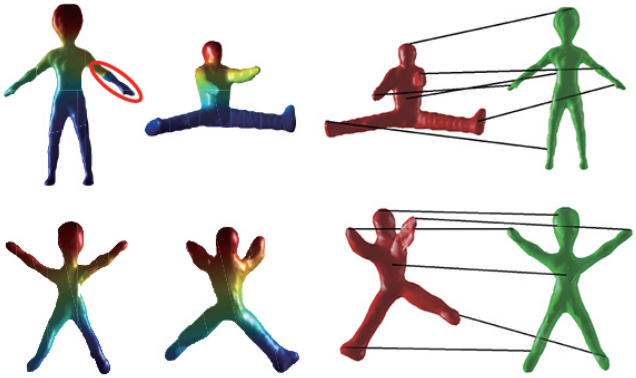

Figure 5: *Matching results. Top: the original meshes and the correspondence. Bottom: the unfolded meshes and the correspondence. Pseudo-color demonstrates the correspondence of mesh1 and mesh2.*

Figure 6: *The thumbnail of PKUNSB models.*

time for a mesh with 1.5k triangle faces is 170s in average. For a mesh with 800 faces, the computational time reduces to 15s on a 3.6GHz PC with 8GB RAM. In our experiments, all meshes are simplified to 1.5k faces.

We evaluate our proposed algorithm in two datasets: SHREC15 [LZC*15] human-like subset and Peking University Non-rigid 3D Shape Benchmark (PKUNSB) dataset [LGX13]. SHREC15 non-rigid 3D shape database† is a 3D shape repository with 1200 meshes belonging to 50 classes. We pick 8 human-like (man, santa, woman, mermaid, mouse, robot, sumotori and woodman) classes from the dataset which consists of 192 models in total. The PKUNSB database contains 90 articulated 3D watertight meshes categorized into 6 classes with equal size. A thumbnail picture of the models in PKUNSB is shown in Figure 6. For visual judgments, we select 8 representative meshes (from 4 different categories) and their corresponding unfolded versions along with the results of 4 other approaches compared, please refer to the supplemental materials for details. Meanwhile, we also illustrate the advantages of our proposed method by two different applications: 3D shape matching and retrieval, where quantitative measurements are provided in shape retrieval tasks.

† http://www.icst.pku.edu.cn/zlian/shrec15-non-rigid/index.htm

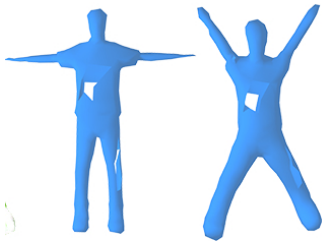

Figure 7: *A mesh with two holes and the resulting unfolded mesh obtained by the proposed method.*

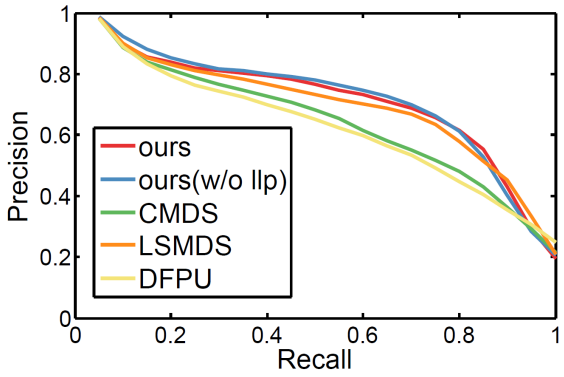

Figure 8: *Precision-recall curve of the retrieval performance evaluated on SHREC15 subset using GSMD descriptors.*

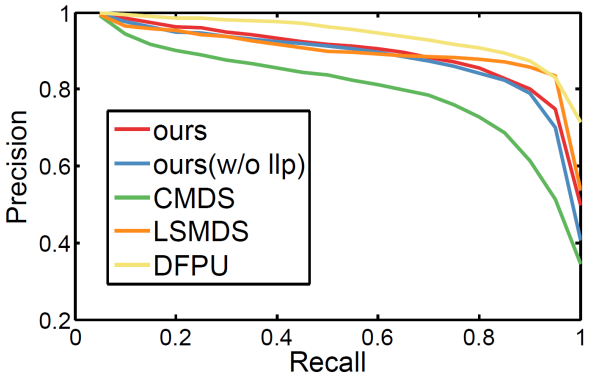

Figure 9: *Precision-recall curve of the retrieval performance evaluated on SHREC15 subset using CMBOF descriptors.*

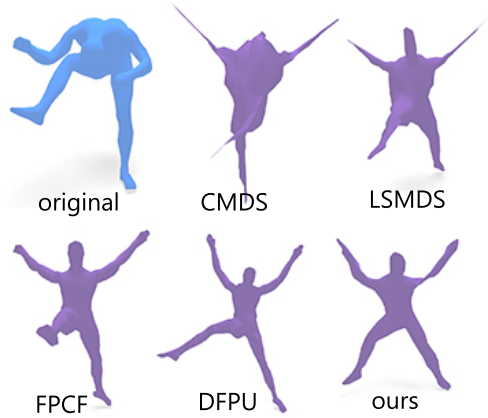

Figure 10: *The stretching intensity of 5 different methods. Top: from left to right, original model, MDS [KW78] and LSMDS [EK03]. Bottom: from left to right, FPCF [LGX13], DFPU [SK16], and our method. The left leg of the human is best unfolded by our method.*

Method	NN	FT	ST	e_Measure	DCG
ours	91.1	**70.8**	**91.2**	**56.9**	**91.1**
ours (w/o LLP)	**93.3**	69.0	90.8	56.5	90.6
CMDS	92.2	65.2	81.7	52.2	88.6
LSMDS	84.4	59.8	77.9	48.8	84.4
DFPU	83.3	58.2	79.7	50.0	83.9
FPCF	85.6	60.4	76.3	48.6	84.7

Table 1: *Retrieval performance evaluated on PKUNSB benchmark using GSMD.*

Method	NN	FT	ST	e_Measure	DCG
ours	**100.0**	82.0	96.8	60.0	96.3
ours (w/o LLP)	97.8	81.3	97.0	59.8	95.8
CMDS	95.5	71.1	91.1	57.0	91.7
LSMDS	97.8	79.4	97.7	60.3	90.0
DFPU	98.9	87.8	97.5	60.0	97.7
FPCF	98.9	**89.0**	**99.4**	**60.7**	**97.3**

Table 2: *Retrieval performance evaluated on PKUNSB benchmark using CMBOF.*

4.2. Shape matching

For point cloud matching, we first sample uniformly on the two meshes to be matched. Then we employ Coherent Point Drift (CPD) [MSC10] algorithm to register 3D point sets. As shown in Figure 5, two human body shapes are sampled and registered. In our implementation, we sample 3k points uniformly on the given meshes. Pseudo-color is utilized to demonstrate the correspondence, i.e. vertices with the same color indicate the pairwise correspondence. 6 salient points are selected for demonstration and the correspondences between two meshes are indicated by lines. Compared to the corresponding results of original shapes, significant improvement has been achieved by matching the unfolded meshes instead. Note that our algorithm operates directly on the original mesh instead of filling the mesh by finite elements, and thus the original topology is preserved without involving additional vertices. Thereby, matching could be performed on the unfolded meshes instead and resulting correspondence can be directly mapped back to the original meshes. As a preprocessing tool, our proposed unfolding method can be integrated into any rigid [SLW02] and non-rigid [BBK*10] shape matching approaches.

4.3. 3D shape retrieval

In this section, we demonstrate the effectiveness of our method in 3D shape retrieval. As we know, the crucial steps of 3D shape retrieval [LGB*13] lie in feature extraction and similarity calculation. In our experiments, we take advantage of two existing view-based features for 3D shape retrieval: the Geodesic Sphere based Multi-view Descriptors (GSMD) [LRS10] and the Clock Matching Bag-of-Features (CMBOF) [LGSX13]. The GSMD serves as a global descriptor while CMBOF is considered as a local feature. For both methods, we normalize the poses of shapes by performing PCA beforehand. For evaluation, we adopt the following five widely-used measures: nearest neighbors, first tier, second tier, e_measure and Discounted Cumulative Gain (DCG). The nearest neighbor calculates the percentage of the closest retrieved shapes that belong to the same class as the query. The first tier calculates the ratio of the related retrieved shapes to the size of query class $|C|$ when the number of the retrieved shapes reaches $|C|$. Second tier share the same meaning as the first-tier except for the number of retrieved shapes reaches $2|C|$. The e_measure indicates the performance of top-ranking positions taking the assumption that results raked higher is more interested. DCG considers the ranking position of the retrieved shapes, i.e. the top-ranked correct results are more weighted than the bottom-ranked ones.

Table 1 and Table 2 show the performance measurements described above for different mesh unfolding methods evaluated in SHREC15 human-like subset using GSMD features and CMBOF features, respectively. Table 3 and Table 4 present the evaluation results in PKUNSB dataset. Figure 8 and Figure 9 depict the Precision-Recall curves of Table 1 and Table 2, respectively. And Figure 11 and Figure 12 plot the PR curves of Table 3 and Table 4. We are unable to execute the FPCF [LGX13] on SHREC15 subset due to the lack of segmentation files.

We can observe from the figures that our proposed method achieves the state-of-the-art performance in GSMD descriptors and

Method	NN	FT	ST	e_Measure	DCG
ours	85.4	**63.4**	77.5	58.7	86.5
ours (w/o LLP)	**89.6**	62.6	**77.8**	**58.8**	**87.4**
CMDS	82.8	54.6	73.3	53.1	83.6
LSMDS	85.4	61.7	77.6	58.1	85.9
DFPU	80.7	53.9	71.8	51.9	82.8

Table 3: *Retrieval performance evaluated on SHREC15 non-rigid subset using GSMD.*

Method	NN	FT	ST	e_Measure	DCG
ours	97.9	84.1	93.1	74.7	95.9
ours (w/o LLP)	97.4	82.4	91.0	73.4	95.0
CMDS	91.7	71.0	89.2	67.5	91.3
LSMDS	94.3	84.2	96.2	76.0	95.3
DFPU	**98.9**	**89.2**	**97.1**	**78.7**	**97.9**

Table 4: *Retrieval performance evaluated on SHREC15 non-rigid subset using CMBOF.*

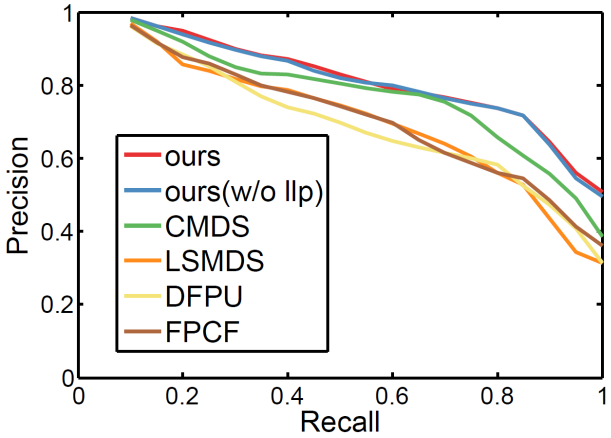

Figure 11: *Precision-recall curve of the retrieval performance evaluated on PKUNSB dataset using GSMD descriptors.*

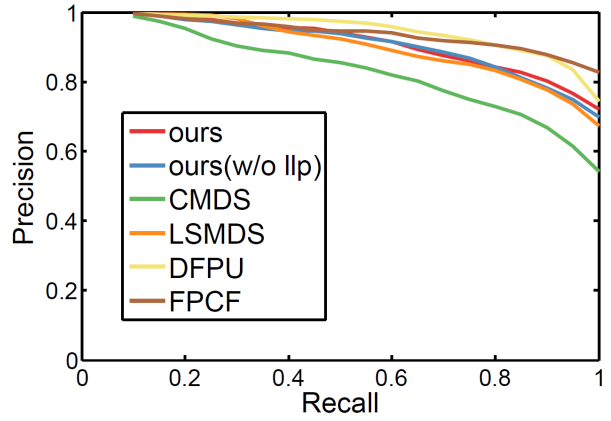

Figure 12: *Precision-recall curve of the retrieval performance evaluated on PKUNSB dataset using CMBOF descriptors.*

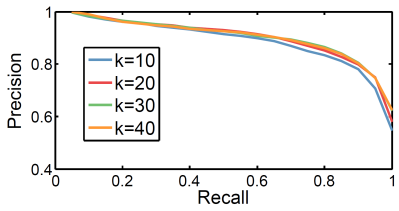

Figure 13: *The Precision-recall curve of choosing different k evaluated on SHREC15 subset using CMBOF features.*

acceptable results in CMBOF features. Yet our method requires no geodesic computation and watertightness for the target mesh. As shown in Figure 7, we add two holes on the original mesh, while the resulting unfolded mesh still keeps plausible. Compared to the existing two feature-preserving unfolding methods [LGX13, SK16], our algorithm requires no tetrahedralization or segmentation, which brings more robustness and usability.

The evaluation results of our unfolding methods with and without locally linear preserving term suggest the necessity of adding this term. As illustrated in Figure 3, the locally linear preserving term significantly improves the visual quality of the unfolded mesh. However, for GSMD features, we do not observe an obviously better performance compared to the simple maximum variance unfolding version. This is probably due to the fact that GSMD captures the global view features which weakens the influence of local details. Contrarily, when utilizing the CMBOF approach, our method with locally linear preserving term outperforms the version with simple maximum variance unfolding. This is mainly due to the fact that CMBOF describes views around the object by extracting SIFT salient points which weights more on detail features. This also explains why our method achieves state-of-the-art performance in GSMD evaluations: for a global descriptor the stretching intensity weights more than the detail-preserving. And for local descriptors this is right on the contrary. This also suggests that our algorithm achieves the strongest stretching intensity with acceptable detail-preserving ability (Figure 10). Therefore, our method is most suitable for global features, which is much faster to obtain than the local descriptors that need storage and clustering. Since the shapes within the dataset are very similar as shown in Figure 6, our method has the potential to be applied to the scenarios where a strong detail-distinguishing ability is expected.

Figure 13 illustrates the influence of choosing different k values. Larger k covers a larger geometry to be preserved while smaller one is on the contrary. We can observe in the figure that when k increases from 10 to 30, the retrieval result improves slightly, but fall back down when k reaches to 40. That is due to the fact that larger k reinforces the detail-preserving but reduces the stretching strength. k is related to the scale of rigid-like component of 3D models. Due to the slightness of influence, we set $k = 30$ across all the shapes in our implementations.

5. Conclusion and limitations

In this paper, we proposed a 3D mesh unfolding algorithm taking advantage of semidefinite programming techniques. By maximiz-

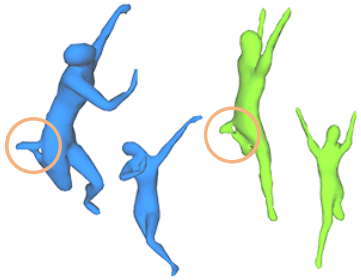

Figure 14: *Failure cases. Left: the original mesh. Right: the unfolded mesh.*

ing the variance of a given mesh, the canonical pose of the mesh is obtained by optimizing a carefully-designed objective function. To obtain more intensity of detail-preserving, we introduced the locally linear preserving term and integrated it into the programming of Gram matrix. We demonstrate the advantages of our method by two applications: 3D shape matching and retrieval. Both quantitative and visual evaluations approve the effectiveness of our algorithm.

Despite those advantages, the proposed method sometimes fails when the topology prevents the mesh to be unfolded. As shown in Figure 14 (note the adhesion between the buttock and foot in human body), our algorithm unfolds the arms and legs successfully, but the adhesion remains unchanged. In these cases, we need to manually cut the adhesion spot and then implement the unfolding algorithm. Also, a certain degree of inflations might be introduced due to the variance maximization. Furthermore, due to the high computational burden of semidefinite programming, our current method can only handle simplified meshes with no more than 1.5k vertices in a common PC. We believe that the problem would be solved as the techniques for SDP progresses. Currently we solve this problem by mapping the simplified mesh back to the mesh with original resolution, which has been demonstrated in the paper (see Figure 3). To unfold meshes directly with high-resolution, landmark-based methods are possible solutions. We leave this as our future work.

ACKNOWLEDGMENTS

This work was supported by National Natural Science Foundation of China (Grant No.: 61472015, 61672056 and 61672043), Beijing Natural Science Foundation (Grant No.: 4152022) and National Language Committee of China (Grant No.: ZDI135-9).

References

[BB13] BARRA V., BIASOTTI S.: 3d shape retrieval using kernels on extended reeb graphs. *Pattern Recognition 46*, 11 (2013), 2985–2999. 2

[BBK*10] BRONSTEIN A. M., BRONSTEIN M. M., KIMMEL R., MAHMOUDI M., SAPIRO G.: A gromov-hausdorff framework with diffusion geometry for topologically-robust non-rigid shape matching. *International Journal of Computer Vision 89*, 2-3 (2010), 266–286. 6

[BG03] BORG I., GROENEN P.: Modern multidimensional scaling: Theory and applications. *Journal of Educational Measurement 40*, 3 (2003), 277âĂŞ280. 2

[EK03] ELAD A., KIMMEL R.: On bending invariant signatures for surfaces. *Pattern Analysis and Machine Intelligence IEEE Transactions on 25*, 10 (2003), 1285–1295. 6

[GB08] GRANT M., BOYD S.: Graph implementations for nonsmooth convex programs. In *Recent Advances in Learning and Control*, Blondel V., Boyd S., Kimura H., (Eds.), Lecture Notes in Control and Information Sciences. Springer-Verlag Limited, 2008, pp. 95–110. http://stanford.edu/~boyd/graph_dcp.html. 4

[GB14] GRANT M., BOYD S.: CVX: Matlab software for disciplined convex programming, version 2.1. http://cvxr.com/cvx, Mar. 2014. 4

[KW78] KRUSKAL J. B., WISH M.: *Multidimensional Scaling.* BOOK ON DEMAND POD, 1978. 6

[LGB*13] LIAN Z., GODIL A., BUSTOS B., DAOUDI M., HERMANS J., KAWAMURA S., KURITA Y., LAVOUÃL' G., NGUYEN H. V., OHBUCHI R.: A comparison of methods for non-rigid 3d shape retrieval. *Pattern Recognition 46*, 1 (2013), 449–461. 6

[LGSX13] LIAN Z., GODIL A., SUN X., XIAO J.: Cm-bof: visual similarity-based 3d shape retrieval using clock matching and bag-of-features. *Machine Vision and Applications 24*, 8 (2013), 1685–1704. 6

[LGX13] LIAN Z., GODIL A., XIAO J.: Feature-preserved 3d canonical form. *International Journal of Computer Vision 102*, 1-3 (2013), 221–238. 2, 5, 6, 7

[LRS10] LIAN Z., ROSIN P. L., SUN X.: Rectilinearity of 3d meshes. *International Journal of Computer Vision 89*, 2-3 (2010), 130–151. 2, 6

[LZC*15] LIAN Z., ZHANG J., CHOI S., ELNAGHY H., ELSANA J., FURUYA T., GIACHETTI A., GULER R. A., LAI L., LI C.: Shrec'15 track: Non-rigid 3d shape retrieval. *Eurographics Workshop Ond Object Retrieval* (2015). 5

[MF11] MA Y., FU Y.: Manifold learning theory and applications. *Crc Press* (2011).

[MSC10] MYRONENKO B. A., SONG X., CARREIRAPERPIÃŚÃąN Ã.: Non-rigid point set registration: Coherent point drift (cpd). *Advances in Neural Information Processing Systems 32*, 12 (2010), 1009–1016. 6

[OOFB08] OHBUCHI R., OSADA K., FURUYA T., BANNO T.: Salient local visual features for shape-based 3d model retrieval. In *IEEE International Conference on Shape Modeling and Applications* (2008), pp. 93–102. 2

[RS01] ROWEIS S. T., SAUL L. K.: Saul, l.: Nonlinear dimensionality reduction by locally linear embedding. science 290, 2323-2326. *Science 290*, 5500 (2001), 2323–6. 4

[Sah15] SAHILLIOÄ§LU Y.: A shape deformation algorithm for constrained multidimensional scaling. *Computers and Graphics 53* (2015). 2

[SAS07] SCOVANNER P., ALI S., SHAH M.: A 3-dimensional sift descriptor and its application to action recognition. In *International Conference on Multimedia* (2007), pp. 357–360. 2

[SK16] SAHILLIOĞLU Y., KAVAN L.: Detail-preserving mesh unfolding for non-rigid shape retrieval. *ACM Trans. Graph. 35*, 2 (2016). 2, 6, 7

[SLW02] SHARP G. C., LEE S. W., WEHE D. K.: Icp registration using invariant features. *IEEE Transactions on Pattern Analysis and Machine Intelligence 24*, 1 (2002), 90–102. 6

[TSL00] TENENBAUM J. B., SILVA V. D., LANGFORD J. C.: A global geometric framework for nonlinear dimensionality reduction. *Science 290*, 5500 (2000), 2319–23. 2

[TV10] TANGELDER J. W. H., VELTKAMP R. C.: A survey of content based 3d shape retrieval methods. In *Proceedings of the Shape Modeling International 2004* (2010), pp. 145–156. 1

[WS04] WEINBERGER K. Q., SAUL L. K.: Unsupervised learning of image manifolds by semidefinite programming. *International Journal of Computer Vision 70*, 1 (2004), 77–90. 2