Eurographics Workshop on 3D Object Retrieval (2015)
I. Pratikakis, M. Spagnuolo, T. Theoharis, L. Van Gool, and R. Veltkamp (Editors)

SHREC'15 Track: Canonical Forms for Non-Rigid 3D Shape Retrieval

David Pickup[1*], Xianfang Sun[1*], Paul L. Rosin[1*], Ralph R. Martin[1*], Zhiquan Cheng[2*], Sipin Nie[3], Longcun Jin[3]

[1]Cardiff University, UK
[2]Avatar Science (Hunan) Company, China
[3]South China University of Technology, China
[*]Track organisers

Abstract

We present a new benchmark for testing algorithms that create canonical forms for use in non-rigid 3D shape retrieval. We have combined two existing datasets to create a varied collection of models for testing. Canonical forms attempt to factor out a shape's pose, giving a pose-neutral shape. This opens up the possibility of using methods originally designed for rigid retrieval for the task of non-rigid shape retrieval. We demonstrate the benchmark by using it to compare the performance of nine canonical form methods, using three different retrieval algorithms.

1. Introduction

The ability to recognise a deformable object's shape, regardless of the pose of the object, is an important requirement for modern shape retrieval methods. One approach to this problem is to transform each deforming model into a canonical form which (ideally) factors out the pose, leaving a standard pose-independent version of the shape. This allows rigid shape retrieval algorithms to be be used for non-rigid shape retrieval. Many different methods for automatically computing a canonical form from a 3D mesh have been proposed. Methods using such approaches along with rigid retrieval systems have performed well on shape retrieval benchmarks [LGB*11]. However, most of these methods have not been compared using the same dataset, or used for retrieval with the same rigid retrieval system, so their relative performance is unclear. We propose a new benchmark to provide a meaningful comparison of existing and new canonical form methods for non-rigid shape retrieval, and use it to make such a comparison. The website for our benchmark, containing the data and source code for evaluation can be found here: http://www.cs.cf.ac.uk/shaperetrieval/shrec15/

2. Datasets

Our benchmark combines some of the models from two existing datasets. The first is the SHREC'11 non-rigid dataset [LGB*11], which contains a wide range of different shape classes. The second is the non-rigid humans dataset [PSR*14], which provides a more challenging test of the performance of the canonical forms, as these shapes only differ in their minor details. Our dataset is split into training and testing sets; it has been kept to a reasonable size to make the tasks listed in Section 3 feasible within the time available.

The largest mesh resolution within the dataset is 60,210 vertices, with an average mesh resolution of 21,141 vertices.

3. Evaluation

Participants in this track were given two tasks.

1. Submit a canonical form for each mesh within the dataset, along with a description of the method used to compute canonical forms.
2. Perform shape retrieval using the canonical forms submitted for Task 1, and submit a description of the method used. The retrieval task is formally defined as

   > Return a list of all models, ordered by decreasing shape similarity to a query model.

Our track had two rounds of submissions. For the first round of submissions, participants submitted their entries for Task 1. All the entries of Task 1 were then sent out to all the participants for Task 2, and the results of this task were collected in the second round of submissions.

DOI: 10.2312/3dor.20151063

The evaluation procedure is similar to that used in several previous SHREC tracks [LGB*11]. We evaluated the retrieval results using various statistical measures: nearest neighbour (NN), first tier (1-T), second tier (2-T), discounted cumulative gain (DCG), and precision and recall curves. Definitions of these measures are given in [SMKF04].

We also discuss the desired qualities of a canonical form. and test the submitted forms for some of these qualities. We then analyse whether there is a correlation between exhibiting the proposed desired qualities, and retrieval accuracy.

4. Canonical Form Methods

We now briefly describe each of the canonical form methods compared in our study.

4.1. Various methods

submitted by D. Pickup, X. Sun, P. L. Rosin, R. R. Martin

As a basis for the study, we entered various canonical form methods, some of our own, and some implemented from the literature. Many are variants of the multidimensional scaling (MDS) and geodesic distance based method proposed by Elad and Kimmel [EK03], but some avoid either or both of these components.

4.1.1. Classical Multidimensional Scaling

Elad and Kimmel [EK03] proposed computing a canonical form of a mesh by mapping the geodesic distance between all pairs of vertices to three-dimensional Euclidean distances. As the geodesic distances are pose invariant, the Euclidean embedding is a pose invariant representation of the shape. We first compute the all-pairs geodesic distance matrix of the mesh using the fast marching method [KS98]. The Euclidean embedding of these distances is then computed using classical MDS. We do this by first calculating

$$B = -\frac{1}{2}JDJ, \tag{1}$$

where

$$J = I - \frac{1}{n}\mathbf{1}\mathbf{1}^T, \tag{2}$$

$$\mathbf{1}_{1xn} = [1,1,...,1]^T, \tag{3}$$

D is the matrix of squared geodesic distances, and n is the number of mesh vertices. We then compute the four largest eigenvalues $(\lambda_0 > \lambda_1 > \lambda_2 > \lambda_3)$ and their associated eigenvectors $(\phi_0, \phi_1, \phi_2, \phi_3)$ of B. Given a point $\mathbf{p}$ on the mesh, the MDS embedding is calculated as

$$\mathrm{MDS}(\mathbf{p}) = \left(\lambda_1^{\frac{1}{2}}\phi_1(\mathbf{p}), \lambda_2^{\frac{1}{2}}\phi_2(\mathbf{p}), \lambda_3^{\frac{1}{2}}\phi_3(\mathbf{p})\right). \tag{4}$$

Given the geodesic distances, the MDS method has a complexity of $O(N^2)$, where N is the number of vertices. However, the computation of geodesic distances has a time complexity of $O(N^2 \log N)$. Due to this high computational expense the meshes are simplified to approximately 2000 vertices before computing the canonical forms. The methods proposed in subsections 4.1.2 – 4.1.4 also need to compute the geodesic distances, so we also simplify the meshes to approximately 2000 vertices in those methods.

4.1.2. Fast Multidimensional Scaling

The fast MDS canonical form method [FL95] uses a variant of the method described in Section 4.1.1, where the geodesic distances are projected into Euclidean space one dimension at a time. We first compute the geodesic distance between each pair of vertices i and j as d_{ij}. For each of the three dimensions in turn we select the two most distant vertices O_a and O_b. All the other vertices are then projected onto the line defined by (O_a, O_b):

$$x_i = \frac{d_{ai}^2 + d_{ab}^2 - d_{bi}^2}{2d_{ab}}. \tag{5}$$

x_i is used as the embedded vertex coordinates in the current dimension. The distances are then updated as

$$d_{i'j'}^2 = d_{ij}^2 - ||x_i - x_j||^2. \tag{6}$$

This MDS method has a lower time complexity of $O(N)$.

4.1.3. Least Squares Multidimensional Scaling

This is another variant on the method described in Section 4.1.1, also proposed by Elad and Kimmel [EK03]. We use the *SMACOF* (scaling by majorizing a complicated function) algorithm to compute the MDS. SMACOF minimises the following functional:

$$S(X) = \sum_{i=1}^{N}\sum_{j=i+1}^{N} w_{i,j}(\delta_{i,j} - d_{i,j}(X))^2, \tag{7}$$

where N is the number of vertices, $w_{i,j}$ are weighting coefficients, $\delta_{i,j}$ is the geodesic distance between vertices i and j of the original mesh, and $d_{i,j}$ is the Euclidean distance between vertices i and j of the resulting canonical mesh X. The stress function is minimised iteratively using the code provided by Bronstein et al. [BBK08].

This MDS algorithm has a complexity of $O(N^2 \cdot \mathrm{NumOfIterations})$.

4.1.4. Non-metric Multidimensional Scaling

This method is very similar to Section 4.1.3, but instead of matching the Euclidean distances to the exact geodesic distances, we only match the ordering of the distances. The stress function we minimise for this method is

$$S(X) = \frac{\sum_{i=1}^{N}\sum_{j=i+1}^{N}(f(\delta_{i,j}) - d_{i,j}(X))^2}{\sum_{i=1}^{N}\sum_{j=i+1}^{N} d_{i,j}(X)^2}, \tag{8}$$

where N is the number of vertices, $\delta_{i,j}$ is the geodesic distance between vertices i and j of the original mesh, $d_{i,j}$ is the

Euclidean distance between vertices i and j of the resulting canonical mesh X, and f is an optimal monotonic function of the dissimilarities. This MDS method is less restrictive than metric MDS, and some researchers have found that it produces more desirable results [KLT05].

4.1.5. Global Point Signatures

We have implemented the method to compute the Global Point Signatures (GPS) embedding of mesh proposed by Rustamov [Rus07]. Firstly we calculate the discrete Laplace-Beltrami operator with cotangent weights [HPW06]. We then compute the four smallest eigenvalues $(\lambda_0 < \lambda_1 < \lambda_2 < \lambda_3)$ and their associated eigenvectors $(\phi_0, \phi_1, \phi_2, \phi_3)$ of the Laplace-Beltrami operator. Given a point $\mathbf{p}$ on the mesh, the GPS embedding is calculated as

$$\mathrm{GPS}(\mathbf{p}) = \left(\frac{1}{\sqrt{\lambda_1}}\phi_1(\mathbf{p}), \frac{1}{\sqrt{\lambda_2}}\phi_2(\mathbf{p}), \frac{1}{\sqrt{\lambda_3}}\phi_3(\mathbf{p}) \right). \quad (9)$$

The eigendecomposition of the Laplace-Beltrami operator is invariant to isometry preserving deformations, and therefore the GPS embedding is a pose invariant representation of the mesh. The GPS embedding is faster to compute than the MDS based method by Elad and Kimmel [EK03] as it avoids the use of geodesic distances. We are therefore able to compute the GPS canonical forms using the full resolution meshes, without any simplification.

4.1.6. Skeleton driven canonical forms

A variant on the canonical forms presented by Elad and Kimmel [EK03] is used. A canonical form is produced by extracting a curve skeleton from a mesh, using the method by Au at al. [ATC*08]. The SMACOF Multidimensional Scaling method used by [EK03] is then applied to the skeleton, to put the skeleton into a canonical pose. The skeleton driven shape deformation method by Yan et al. [YHMY08] is then used to deform the mesh to the new pose defined by the canonical skeleton. This produces a similar canonical form to [EK03], but with the local features better preserved. The models in the dataset are simplified to approximately 15000 vertices before computing the canonical forms.

4.1.7. Euclidean Distance Based Canonical Forms

We compute a canonical form of a mesh by stretching out its limbs so that its extremities are distant from one another. We achieve this effect more efficiently than the common method of computing multidimensional scaling (MDS) on the geodesic distances [EK03]. We instead maximise the Euclidean distances between feature points on the extremities of the mesh, whilst preserving the original edge lengths to ensure isometric deformation. Our method has been accepted for publication [PSRM15].

We first scale the mesh so that the maximum distance of any point on its surface to the centroid of all vertices is one. We then use the method of Ben-Chen and Gotsman [BCG08] to calculate the conformal factor of the mesh. The conformal factor increases along the length of mesh protrusions, which results in high values at the extremities of the mesh. To obtain a set of feature points for a mesh with N vertices, we sample the $\sqrt{N}$ vertices which have the largest conformal factors and also satisfy the requirement that they are local maxima. A vertex is defined to be a local maximum if its conformal factor is greater than that of all its neighbours in a 2-ring neighbourhood. We select $\sqrt{N}$ feature points, as this is the largest number of features we can have whilst being able to compute the Euclidean distances between all pairs of feature points in linear time (with respect to the number of mesh vertices).

We compute the canonical form of the mesh by adapting the least-squares MDS formulation used by [EK03]

$$S(X) = \sum_{i=1}^{N} \sum_{j=i+1}^{N} w_{i,j}(\delta_{i,j} - d_{i,j}(X))^2, \quad (10)$$

where N is the number of vertices, $w_{i,j}$ are weighting coefficients, and $d_{i,j}$ is the Euclidean distance between vertices i and j of the resulting canonical mesh X. We set the value of $\delta_{i,j}$ for all connected vertices i and j equal to the length of the edge connecting them. This aims to preserve the edge lengths of the mesh, to ensure isometric deformation. In order to maximise the distance between feature points, the value of $\delta_{i,j}$ for each pair of the $\sqrt{N}$ sampled vertices is set to a high value α. We want this value to be large enough to straighten all the limbs of the model, and our experiments show 10 is large enough. As long as α is large enough and the parameter β discussed below is optimised accordingly, any value of α can be chosen.

If the two vertices i and j are neither a pair of feature points nor connected by an edge, we do not enforce a target distance between them, so $w_{i,j}$ is set to zero for such cases. Not having to compute and optimise the distances between these points is crucial in keeping the linear time complexity of our distance calculations. The weights $w_{i,j}$ in Equation 10 for all i and j that are connected by an edge are set to $\beta/\delta_{i,j}^2$, where β is a user defined parameter for preserving edge lengths. We divide by the square of the edge length $\delta_{i,j}^2$ so that the distance in Equation 10 becomes a relative, rather than absolute, difference, making the weighting independent of the length of the edge. The conformal factor is normalised to lie in the interval $[0,1]$, and the entries in the weighting matrix $w_{i,j}$ for each pair of feature points are set to the mean of their conformal factors. This results in vertices which are nearer the ends of long 'limbs' of the object having a higher impact on the resulting canonical form, and avoids stretching out inappropriate parts of the mesh. The SMACOF algorithm can then be used to minimise Equation 10 as described in [EK03].

In many cases the number of local maxima of the conformal factor is less than $\sqrt{N}$. We want the number of feature

points to be exactly $\sqrt{N}$ so that the number of edges connecting pairs of feature points grows at the same rate as the number of mesh vertices. This in turn ensures that we can use the same value for the parameter β regardless of mesh resolution. We offer two different solutions to handling this issue. The first is to increase the number of feature points to $\sqrt{N}$ by adding extra randomly selected vertices as feature points. We refer to this method as "Euclidean Random" in Section 6.

The second is to separately normalise the weightings $w_{i,j}$ used for pairs of feature points, and for adjacent vertices. We normalise the weights for adjacent vertices by dividing by the total number of edges, and we normalise the feature point pair weights by dividing by the sum of all feature point pair weights. Thus, we may rewrite the final functional to be minimised as

$$S(X) = \sum_{i\in F}\sum_{j\in F, j\neq i} \frac{0.5(\Phi_i+\Phi_j)}{\sum_{i\in F}\sum_{j\in F, j\neq i} 0.5(\Phi_i+\Phi_j)}(\delta_{i,j}-d_{i,j}(X))^2 + \sum_{(i,j)\in E}\frac{\beta}{|E|\delta_{i,j}^2}(\delta_{i,j}-d_{i,j}(X))^2, \tag{11}$$

where F is the set of all feature points, E is the set of all edges, and Φ_i is the conformal factor of vertex i. We refer to this second method as "Euclidean Normalised" in Section 6.

This method eliminates the need for the expensive geodesic distance computation. The feature point selection and Euclidean distance calculations have a time complexity of $O(N)$, and the distance computations for each iteration of the MDS stress minimisation algorithm are also computed in $O(N)$ time. We are therefore able to compute these canonical forms using the full resolution meshes.

4.2. Least Squares MDS B
submitted by S.-P. Nie, L.-C. Jin

This method is identical to the Least Squares MDS method described in Section 4.1.3, but with one minor alteration. Normally at the end of each iteration of the SMACOF algorithm the change in the stress function (Equation 7) is measured, and if the change is less than a small value ε the algorithm terminates. In this method we instead measure the change in the following sum of absolute differences between the original geodesic and current Euclidean distances:

$$S(X) = \sum_{i=1}^{N}\sum_{j=i+1}^{N} w_{i,j}|\delta_{i,j}-d_{i,j}(X)|. \tag{12}$$

This results in an earlier termination of the algorithm.

This method retains the $O(N^2 \log N)$ time complexity, as it still requires the use of geodesic distances. We simplify the meshes to approximately 2500 vertices before computing the canonical forms.

5. Retrieval Methods

We now briefly describe each of the retrieval methods compared in our study.

5.1. CM-BOF and 3DSP
submitted by D. Pickup, X. Sun, P. L. Rosin, R. R. Martin

We use two local feature based methods from existing literature to produce retrieval results, one based on extracting features from rendered depth maps of the shape (Section 5.1.1), and another based on extracting features from the 3D mesh itself (Section 5.1.2).

5.1.1. Clock Matching and Bag-of-Features

The Clock Matching and Bag-of-Features (CM-BOF) method was proposed by Lian et al. [LGSX13], and has been shown to perform extremely well at non-rigid 3D shape retrieval when combined with a canonical form method [LGB*11]. To compute a descriptor of a 3D shape we first centre the mesh, normalise its scale, and use a combination of Principle Component Analysis (PCA) and Rectilinearity [LRS10] to normalise its orientation. We then render 66 depth images of the mesh from viewpoints on the vertices of a geodesic sphere. SIFT features [Low04] are then extracted from the depth images. A 1000 length histogram descriptor of each image of the shape is then created using the bag-of-words method. The similarity of two shapes is calculated as the sum of the similarities of their matching views.

5.1.2. 3D Spatial Pyramids

We use the 3D Spatial Pyramids (3DSP) method proposed by Redondo-Cabrera et al. [RCLSARMB14]. To describe a shape, we first extract a set of 3D SURF features [KPW*10] from the 3D mesh itself. The volume containing the mesh is recursively broken down into sub-cubes, forming a spatial pyramid. A bag-of-words is then computed for each part of the pyramid and then the histograms are concatenated to form the shape's final descriptor. The similarity of two shapes is computed as the similarity of their descriptors.

5.2. Histogram of Mean Curvature
submitted by S.-P. Nie, L.-C. Jin

To produce a shape's descriptor we first use the code made available by Dirk-Jan Kroon [Kro] to compute the mean curvature for every vertex of the mesh. This code computes the mean curvature for a mesh vertex by first rotating the mesh so that the vertex's normal lies in the direction $(-1,0,0)$, and the local neighbourhood is then treated as two-dimensional. A quadratic patch is then fitted to this local neighbourhood: $f(x,y) = ax^2+by^2+cxy+dx+ey+f$. The eigenvectors and eigenvalues of the Hessian are then used to calculate the mean curvature. We generate a 64 bin histogram of the mean curvatures, and normalise it to sum to 1000. The similarity

between two shapes is calculated as the similarity between their curvature histograms.

This method fails on the canonical forms produced by the GPS method (Section 4.1.5).

6. Results

We divide this section into two parts. First we look at the properties of the canonical forms themselves in Section 6.1, and then we evaluate the retrieval results in Section 6.2.

6.1. Canonical Forms

Objects which have the same shape, but are in different poses, should have identical canonical forms. Objects which differ in shape should have different canonical forms, even if they are in a similar pose. Therefore a canonical form should remove any traces of the pose of the original shape, but preserve the features of the shape which are pose invariant.

Figure 1 shows a selection of meshes and their canonical forms produced by each of the participating methods. The common feature we see in these canonical forms is that the mesh is "straightened out" and the extremities are placed distant from one another. The different methods vary in degree of straightening, and how much high frequency detail from the original shape is preserved. The two variants of the Euclidean distance based method (Section 4.1.7) allow an explicit trade-off between these two properties by adjusting a weight.

We have attempted to quantify how much the canonical forms distort the mesh's original shape, and the consistency of the different canonical forms of the same shape in different poses. To quantify the amount of distortion of the original shape, we calculate the difference in the compactness measure between the original mesh and the canonical form (Table 1). The compactness measure is calculated as V^2/A^3, where V is the mesh volume and A is the mesh surface area. To quantify the consistency of the canonical forms across models of the same shape, we use iterative closest point matching to align each pair of shapes of the same class and compute their Haussdorf distance using the Metro software [CRS98] (Table 2).

The Skeletons method submitted by Pickup et al. has the smallest error using the compactness measure, suggesting it is best at preserving the shape of the original model. Figure 1 shows that this method does appear to preserve much of the local detail of the mesh, especially relative to many of the other methods. It is interesting to note however, that the Skeletons method has the largest within-class Haussdorf distance, indicating that the canonical forms are not as consistent as the other methods. The two variants of the Euclidean method by Pickup et al. also perform better than most methods at preserving compactness, but worse than most methods at producing a small Haussdorf distance. These results may indicate that there is a trade-off between producing a consistent canonical form, and preserving the details of the shape.

	CM-BOF	3DSP	Curvature
Original Meshes	0.50	0.47	0.16
Classic MDS	0.73	0.56	0.10
Fast MDS	0.66	0.57	0.14
Least Squares MDS	0.75	0.69	0.11
Non-Metric MDS	0.77	0.68	0.10
GPS	0.72	0.50	—
Skeletons	0.74	0.66	0.15
Euclidean Random	0.54	0.56	0.07
Euclidean Normalised	0.61	0.58	0.18
Least Squares MDS B	0.66	0.60	0.04

Table 3: *Nearest neighbour.*

	CM-BOF	3DSP	Curvature
Original Meshes	0.567	0.451	0.098
Classic MDS	0.597	0.366	0.104
Fast MDS	0.590	0.332	0.110
Least Squares MDS	0.694	0.468	0.103
Non-Metric MDS	0.687	0.478	0.099
GPS	0.556	0.368	—
Skeletons	0.682	0.462	0.101
Euclidean Random	0.640	0.469	0.069
Euclidean Normalised	0.673	0.468	0.108
Least Squares MDS B	0.662	0.454	0.087

Table 4: *First tier.*

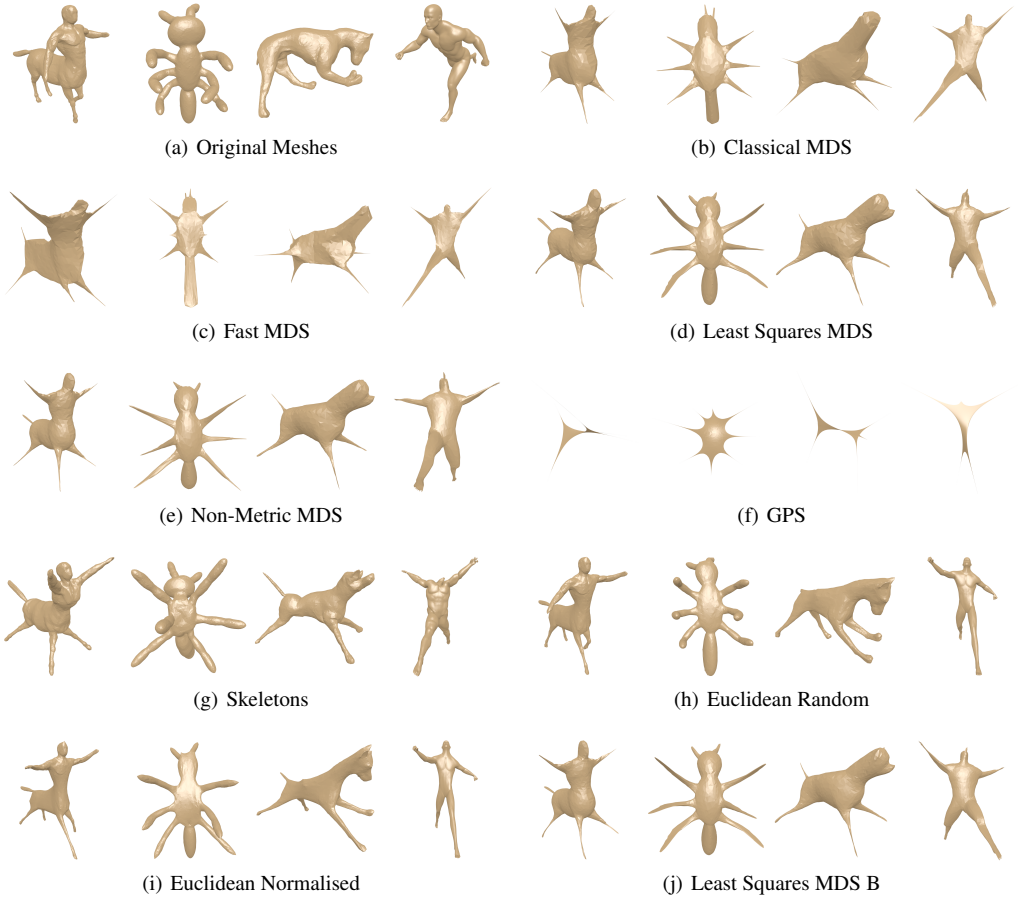
6.2. Retrieval

Tables 3–6 show the retrieval accuracy of the three different retrieval methods on each set of canonical forms, using the nearest neighbour (NN), first tier (FT), second tier (ST) and discounted cumulative gain (DCG) measures. Figure 2 shows the precision-recall (PR) curves for each of the three retrieval algorithms on the different sets of canonical forms.

For the CM-BOF method submitted by Pickup et al., all the canonical forms, except the GPS canonical forms, per-

	CM-BOF	3DSP	Curvature
Original Meshes	0.702	0.610	0.199
Classic MDS	0.741	0.532	0.178
Fast MDS	0.718	0.482	0.183
Least Squares MDS	0.829	0.622	0.189
Non-Metric MDS	0.811	0.640	0.188
GPS	0.697	0.546	—
Skeletons	0.791	0.602	0.206
Euclidean Random	0.783	0.644	0.142
Euclidean Normalised	0.796	0.637	0.202
Least Squares MDS B	0.788	0.620	0.172

Table 5: *Second tier.*



(a) Original Meshes (b) Classical MDS

(c) Fast MDS (d) Least Squares MDS

(e) Non-Metric MDS (f) GPS

(g) Skeletons (h) Euclidean Random

(i) Euclidean Normalised (j) Least Squares MDS B

Figure 1: *Example canonical forms produced by each method.*

	Classical MDS	Fast MDS	Least Squares MDS	Non-Metric MDS	GPS	Skeletons	Euclidean Random	Euclidean Normalised	1-Form MDS
Mean Error	8.44	6.60	4.77	5.17	9.38	0.71	1.86	2.97	4.89
Standard Deviation	4.420	3.384	3.595	3.754	2.641	0.907	1.362	1.231	3.712

Table 1: *Error in compactness between the canonical forms and the original meshes. The values have been multiplied by* 10^4.

	Classical MDS	Fast MDS	Least Squares MDS	Non-Metric MDS	GPS	Skeletons	Euclidean Random	Euclidean Normalised	1-Form MDS
Mean Error	0.339	0.393	0.346	0.276	0.460	0.543	0.463	0.469	0.322
Standard Deviation	0.243	0.227	0.183	0.155	0.378	0.835	0.308	0.193	0.170

Table 2: *Haussdorff distances between canonical forms of the same class.*

	CM-BOF	3DSP	Curvature
Original Meshes	0.753	0.684	0.422
Classic MDS	0.796	0.653	0.402
Fast MDS	0.789	0.640	0.414
Least Squares MDS	0.838	0.739	0.410
Non-Metric MDS	0.836	0.749	0.403
GPS	0.783	0.646	—
Skeletons	0.825	0.710	0.409
Euclidean Random	0.793	0.714	0.382
Euclidean Normalised	0.816	0.717	0.427
Least Squares MDS B	0.813	0.701	0.388

Table 6: *Discounted cumulative gain.*

formed better than using the original meshes. This highlights the advantage of using canonical forms for non-rigid retrieval. The PR curves show that the GPS canonical forms improved the precision for low recall values, but caused a worse performance for higher recall values. The GPS canonical forms are outperformed by the original meshes for both the other retrieval methods as well, indicating that these canonical forms may be better suited to other applications than retrieval, such as segmentation.

When using the 3DSP retrieval method submitted by Pickup et al., both the Fast and Classic MDS methods cause a worse performance than when using the original meshes. This shows that these canonical forms are not suitable to achieve an increase in performance when using 3DSP.

For the Curvature retrieval method submitted by Nie et al., using the original meshes outperformed many of the canonical forms. The Euclidean Normalised method by Pickup et al. was the only method to outperform the original meshes on all of the performance measures. This shows that the Curvature method may not be suitable for use with canonical forms. This is also the worst performing retrieval method we tested overall.

Overall the GPS, Fast MDS and Classic MDS methods are outperformed by the other canonical form methods. These three canonical forms have the highest error in preserving compactness (Table 1), which may indicate that they cause too much distortion to the mesh's shape. The rest of canonical form methods result in very similar retrieval accuracies. The Least Squares MDS and Non-Metric MDS methods submitted by Pickup et al. both outperformed all other canonical forms when using CM-BOF retrieval, and performed amongst the top methods when using 3DSP retrieval.

All the MDS based methods which use geodesic distances had to simplify the meshes to either 2000 or 2500 vertices due to their high computational complexity. The Skeleton method only required simplification to 15000 vertices to perform the computation within a reasonable time. Only the Euclidean methods and the GPS method are efficient enough to use the full resolution meshes. The Euclidean methods demonstrate a competitive retrieval performance with a significant increase in efficiency.

7. Conclusion

This paper compared the canonical forms produced by nine different methods. This is the first attempt at comparing a range of different canonical forms for non-rigid shape retrieval. We used three different retrieval methods to evaluate the advantages of using the canonical forms for retrieval, avoiding any biases caused by a single retrieval method.

We have found that the best canonical form methods produce very similar performance, and it is therefore difficult to rank their effectiveness. Some of the older MDS based methods [EK03] are still as effective as the more recent algorithms submitted, but have been surpassed in terms of efficiency. The choice of which algorithm is better is therefore still questionable, and dependent upon the individual requirements of each user.

Acknowledgements

This work was supported by EPSRC Research Grant EP/J02211X/1, National Natural Science Foundation of China Grant 61300135, Hong Kong Scholars Program Grant XJ2014058, Doctoral Fund of Ministry of Education of China Grant 20130172120001, Natural Science Foundation of Guangdong Province Grant S2013040016930, and Open Research Fund of State Key Laboratory Grant I3I03.

References

[ATC*08] AU O. K.-C., TAI C.-L., CHU H.-K., COHEN-OR D., LEE T.-Y.: Skeleton extraction by mesh contraction. SIGGRAPH '08, ACM, pp. 44:1–44:10. 3

[BBK08] BRONSTEIN A. M., BRONSTEIN M. M., KIMMEL R.: *Numerical geometry of non-rigid shapes*. Monographs in Computer Science. Springer, 2008. 2

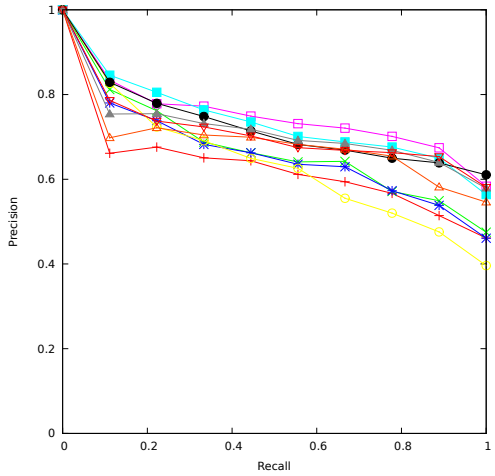
[BCG08] BEN-CHEN M., GOTSMAN C.: Characterizing shape using conformal factors. In *Proceedings of the 1st Eurographics conference on 3D Object Retrieval* (2008), EG 3DOR'08, Eurographics Association, pp. 1–8. 3

[CRS98] CIGNONI P., ROCCHINI C., SCOPIGNO R.: Metro: Measuring error on simplified surfaces. *Computer Graphics Forum 17*, 2 (1998), 167–174. 5

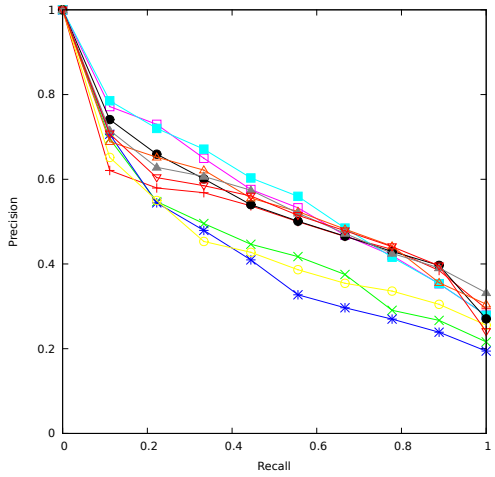
[EK03] ELAD A., KIMMEL R.: On bending invariant signatures for surfaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence 25*, 10 (2003), 1285–1295. 2, 3, 7

[FL95] FALOUTSOS C., LIN K.-I.: Fastmap: A fast algorithm for indexing, data-mining and visualization of traditional and multimedia datasets. In *Proceedings of the 1995 ACM SIGMOD International Conference on Management of Data* (New York, NY, USA, 1995), SIGMOD '95, ACM, pp. 163–174. 2

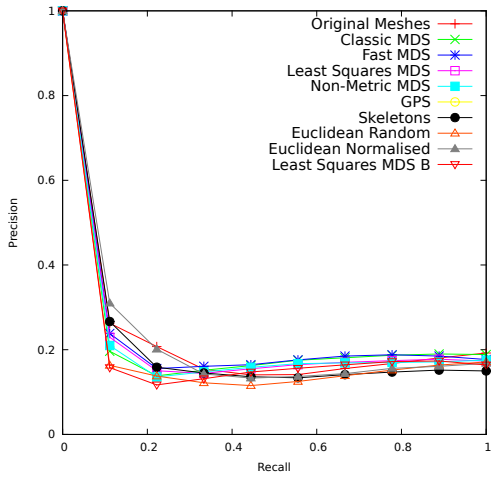
[HPW06] HILDEBRANDT K., POLTHIER K., WARDETZKY M.: On the convergence of metric and geometric properties of polyhedral surfaces. *Geometriae Dedicata 123*, 1 (2006), 89–112. 3



(a) CM-BOF



(b) 3DSP



(c) Curvature

Figure 2: *Precision-Recall curves.*

[KLT05] KATZ S., LEIFMAN G., TAL A.: Mesh segmentation using feature point and core extraction. *The Visual Computer 21*, 8-10 (2005), 649–658. 3

[KPW*10] KNOPP J., PRASAD M., WILLEMS G., TIMOFTE R., VAN GOOL L.: Hough transform and 3d surf for robust three dimensional classification. In *Proceedings of the 11th European Conference on Computer Vision: Part VI* (Berlin, Heidelberg, 2010), ECCV'10, Springer-Verlag, pp. 589–602. 4

[Kro] KROON D.-J.: Patch curvature. http://uk.mathworks.com/matlabcentral/fileexchange/32573-patch-curvature. Accessed: February 2015. 4

[KS98] KIMMEL R., SETHIAN J. A.: Computing geodesic paths on manifolds. *Proceedings of the National Academy of Sciences 95*, 15 (1998), 8431–8435. 2

[LGB*11] LIAN Z., GODIL A., BUSTOS B., DAOUDI M., HERMANS J., KAWAMURA S., KURITA Y., LAVOUÉ G., NGUYEN H. V., OHBUCHI R., OHKITA Y., OHISHI Y., PORIKLI F., REUTER M., SIPIRAN I., SMEETS D., SUETENS P., TABIA H., VANDERMEULEN D.: SHREC'11 track: shape retrieval on non-rigid 3d watertight meshes. In *Proceedings of the 4th Eurographics conference on 3D Object Retrieval* (2011), EG 3DOR'11, Eurographics Association, pp. 79–88. 1, 2, 4

[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* (2013), 1–20. 4

[Low04] LOWE D.: Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision 60*, 2 (2004), 91–110. 4

[LRS10] LIAN Z., ROSIN P., SUN X.: Rectilinearity of 3d meshes. *International Journal of Computer Vision 89*, 2-3 (2010), 130–151. 4

[PSR*14] PICKUP D., SUN X., ROSIN P. L., MARTIN R. R., CHENG Z., LIAN Z., AONO M., BEN HAMZA A., BRONSTEIN A., BRONSTEIN M., BU S., CASTELLANI U., CHENG S., GARRO V., GIACHETTI A., GODIL A., HAN J., JOHAN H., LAI L., LI B., LI C., LI H., LITMAN R., LIU X., LIU Z., LU Y., TATSUMA A., YE J.: SHREC'14 track: Shape retrieval of non-rigid 3d human models. In *Proceedings of the 7th Eurographics workshop on 3D Object Retrieval* (2014), EG 3DOR'14, Eurographics Association. 1

[PSRM15] PICKUP D., SUN X., ROSIN P. L., MARTIN R. R.: Euclidean-distance-based canonical forms for non-rigid 3d shape retrieval. *Pattern Recognition* (2015). 3

[RCLSARMB14] REDONDO-CABRERA C., LÓPEZ-SASTRE R. J., ACEVEDO-RODRÍGUEZ J., MALDONADO-BASCÓN S.: Recognizing in the depth: Selective 3d spatial pyramid matching kernel for object and scene categorization. *Image and Vision Computing*, 32 (2014), 965–978. 4

[Rus07] RUSTAMOV R. M.: Laplace-beltrami eigenfunctions for deformation invariant shape representation. In *Proceedings of the 5th Eurographics symposium on Geometry processing* (2007), Eurographics Association, pp. 225–233. 3

[SMKF04] SHILANE P., MIN P., KAZHDAN M., FUNKHOUSER T.: The princeton shape benchmark. In *Proceedings of Shape Modeling Applications.* (2004), pp. 167–178. 2

[YHMY08] YAN H.-B., HU S.-M., MARTIN R., YANG Y.-L.: Shape deformation using a skeleton to drive simplex transformations. *Visualization and Computer Graphics, IEEE Transactions on 14*, 3 (2008), 693–706. 3