

Local Feature-Based 3D Canonical Form

Hela Haj Mohamed¹ (✉), Samir Belaid¹, and Wady Naanaa²

¹ FSM, University of Monastir, Monastir, Tunisia
hajmohamedhela@yahoo.fr, Samir.Belaid@fsm.rnu.tn

² ENIT, El-Manar University, Tunis, Tunisia
wady.naanaa@gmail.com

Abstract. In this paper, we present a novel approach to compute 3D canonical forms which is useful for non-rigid 3D shape retrieval. We resort to using the feature space to get a compact representation of points in a small-dimensional Euclidean space. Our aim is to improve the classical Multi-Dimensional Scaling MDS algorithm to avoid the super-quadratic computational complexity. To this end, we compute the canonical form of the local geodesic distance matrix between pairs of a small subset of vertices in local feature patches. To preserve local shape details, we drive the mesh deformation by the local weighted commute time. When used as a spatial relationship between local features, the invariant properties of the Biharmonic distance improve the final results.

We evaluate the performance of our method by using two different measures: the compactness measure and the Hausdorff distance.

Keywords: 3D canonical forms · Multidimensional scaling · 3D non-rigid shape · Biharmonic distance

1 Introduction

In the last few years, the recognition task of non-rigid 3D shapes, invariant to object's pose, becomes a significant challenge for modern shape retrieval methods. The review of algorithms for non-rigid 3D shape retrieval is mainly classified into algorithms employing local features, topological structures, isometric-invariant global geometric properties, direct shape matching, and canonical forms. In the last category, many authors proposed to transform each deformable model into a canonical form invariant to the pose. This proposal allows the rigid shape descriptors to be used in non-rigid shape retrieval.

The Multidimensional Scaling Method (MDS) is one of the most used methods for computing 3D canonical forms. However, the main challenge of the MDS based methods remains the construction of canonical forms with well-preserved features and with low time-complexity computation.

This paper introduces a novel canonical form based on multidimensional scaling (MDS) by considering local interest points. Our main idea amounts to partitioning the global MDS problem into a set of sub-problems and then to

generating local 3D canonical forms of the original one. More specifically, feature points of the target model are first detected and then a set of local patches is generated. Thereafter, these sub-parts are transformed into their 3D canonical sub-forms by solving nonlinear minimization problems. Assuming this solution enables the reduction of the high computational cost of geodesic distance between each pair of vertices. Eventually, a spatial relationship constraint is needed between partitions to obtain the final 3D canonical form.

The rest of the paper is organized as follows: In Sect. 2, we briefly present MDS-based techniques. In Sect. 3, we establish the mathematical background of the proposed method. Then, in Sects. 4 and 5, we detail our method and report our experimental results.

2 MDS-Based Techniques

MDS is widely considered as an efficient tool to solve MDS problems. The basic idea of MDS-based algorithms is to map the dissimilarity measure between a pair of features, described in an initial feature space, into the distance between the corresponding pair of points in a small-dimensional Euclidean space. In fact, MDS maps each feature $Y_i, i = 1, \dots, N$ to its corresponding point $X_i, i = 1, \dots, N$ in a m -dimensional Euclidean space R^m by minimizing a given stress function:

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

where $d_F(Y_i, Y_j)$ is the dissimilarity between features Y_i and Y_j , $d_E(X_i, X_j)$ is the Euclidean distance between points X_i and X_j in R^m , and $w_{i,j}$ is a weighting coefficient. The difference between $d_F(Y_i, Y_j)$ and $d_E(X_i, X_j)$ plays the role of the objective function, (also designated by *stress function*), $E_S(X)$ to be minimized. To solve this non-convex minimization problem, many algorithms have already been proposed. As standard optimization algorithm, we may mention Classical MDS [1], Least Squares MDS [2], Fast MDS [3], Non-metric MDS [4], etc.

In 3D domain, the first computed canonical form is performed by Elad and Kimmel [5]. The authors proposed using the least squares multidimensional scaling to generate a canonical form for a given 3D mesh. Wherein the dissimilarity measure between features is calculated by the geodesic distances. To improve the quality of canonical forms, Lian et al. [6] created a feature preserving canonical form by considering MDS embedding results as references and then naturally deform the original meshes against them. Nevertheless, this method is sensitive to topological errors, although it is quite robust against mesh segmentation results. In addition, the methods suffer from the high computational time due to the computation of geodesic distances between all pairs of vertices. To circumvent this difficulty, meshes are approximately simplified to 2000 vertices before the MDS embedding procedure is applied. Nevertheless, this solution can affect the quality of the mesh as well as its local features.

More recent methods are based on local feature [7, 8]. For instance, Pickup et al. [8] have suggested a linear time complexity method for computing a canonical form. The authors maximized the distance between pairs of detected feature points whilst preserving the mesh’s edge lengths. This is achieved by using Euclidean distances between pairs of a small subset of vertices. The same authors, proposed to perform the unbending on the skeleton of the mesh, and use this to drive the deformation of the mesh itself. They successfully saved computational time, and reduced distortion of local shape detail. However, this method is sensible to topological errors that may corrupt the mesh.

3 Mathematical Background

The Classical MDS, proposed by Elad and Kimmel [5] is based on a square and symmetric distance matrix D_F defined by

$$D_F = \begin{bmatrix} d_F^2(Y_1, Y_1) & \dots & d_F^2(Y_1, Y_N) \\ \vdots & \dots & \vdots \\ d_F^2(Y_N, Y_1) & \dots & d_F^2(Y_N, Y_N) \end{bmatrix} \quad (2)$$

where $d_F(Y_i, Y_j)$ is the geodesic distance between the pair Y_i and Y_j , computed using the fast marching method [15] in the feature space. The inner product matrix (i.e., the Gram matrix) G_E is calculated in the embedded Euclidean space by

$$G_E = -\frac{1}{2}JD_FJ \quad (3)$$

where

$$J = \frac{1}{N}\mathbf{1}\mathbf{1}^T \quad (4)$$

where $\mathbf{1}$ denotes the N -one vector.

The Euclidean embedding of these distances is then computed using the eigen-decomposition of the Gram matrix G_E . So, the classical MDS minimizes the following energy function, instead of the stress function $E_S(X)$:

$$E_{S1}(X) = \|Q(\Lambda - \bar{\Lambda})Q^T\|^2 \quad (5)$$

where $\|\bullet\|$ denotes the Frobenius norm of the squared matrix elements, $\Lambda_{m \times m} = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_m)$ are eigenvalues of G_E ordered so that $\lambda_1 \geq \lambda_2 \geq \dots \lambda_k \geq 0$, Q denotes the matrix having as columns the corresponding eigenvectors and m is the dimension of the embedded Euclidean space.

The same authors [5] improved their results by suggesting a standard optimization algorithm. The Least Squares technique uses the SMACOF (Scaling by Maximizing a Convex Function) (Borg and Groenen [2]) algorithm to minimize the following stress function $E_S(X)$:

$$E_S(X) = \sum_{i=1}^N \sum_{j=i+1}^N \omega_{i,j} (d_F(Y_i, Y_j) - d_E(X_i, X_j))^2 \quad (6)$$

where N is the number of vertices, the $w_{i,j}$'s are weighting coefficients, $d_F(Y_i, Y_j)$ is the geodesic distance between vertices Y_i and Y_j in the original mesh, and $d_E(X_i, X_j)$ is the Euclidean distance between vertices X_i and X_j of the resulting canonical mesh X . The algorithm iterates until $|S(X_i) - S(X_{i+1})|$ is less than a user-defined threshold ϵ . This MDS algorithm has a complexity of $\mathcal{O}(N^2 \times \text{NumOfIterations})$.

In the present paper, we adopt the Least Squares technique and the SMA-COF algorithm to compute the 3D canonical form. Our solution decreases the computational cost of geodesic distance computation, by using a set of local dissimilarity matrix.

4 Our Contribution

The main idea behind our method is to construct canonical forms that are invariant to the pose of 3D shapes. It is easy to see that the MDS embedding results naturally deform original deformable parts to normalised poses. For this reason, we resort to calculating local canonical form for each salient patch. We start by detecting the limbs of a given model using an automatic and unsupervised 3D salient point detector. The later task is based on the Auto Diffusion Function (ADF) proposed in [10]. The scalar function ADF is defined on the mesh surface as:

$$ADF_t = K(x, x, \frac{t}{\lambda_1}) = \sum_{i=0} e^{-t \frac{\lambda_i}{\lambda_1}} h_i^2(x) \quad (7)$$

where λ and h are eigenfunctions of Laplace-Beltrami operator (LBO). ADF function is controlled by a single parameter t which can be interpreted as feature scale. The local maximum of the ADF is proved to be the natural feature points of the shape. A demonstration of the invariance of extracted points to non-rigid transformation, scaling, occultation and their insensitive to noise is given in [11]. Figure 1 illustrates examples of feature points extraction based on the ADF function. The parameter t was fixed to 0.1.

The next step is to partition the 3D mesh into local regions using a Voronoi diagram, where seeds are the set of feature points. We then compute the canonical form of the mesh by calculating the value of $\delta_{i,j}$, for all vertices in the same patch. The value of $\delta_{i,j}$ is equal to the length of the shortest path connecting vertex i to vertex j . This aims to minimising the computation cost of the geodesic distance and avoid exploring all the paths between every two points in the original space. Thus, the estimated complexity of this algorithm is $\mathcal{O}(km_k^2)$, where k denotes the number of local patches and m_k is the number of vertices in the k^{th} region. In order to speed up the computation of geodesic distances, we use the heat method proposed by Crane et al. in [13], which gives an approximation of the geodesic distance by exploiting heat kernels. Figure 2 illustrates the results obtained by applying the fast marching algorithm and heat geodesic method. In this work, we used the Matlab source code available on the web site of the book (Bronstein et al. [14]) to calculate the canonical form. Clearly, the heat method gives similar results of the embedding form and has high convergence speed.

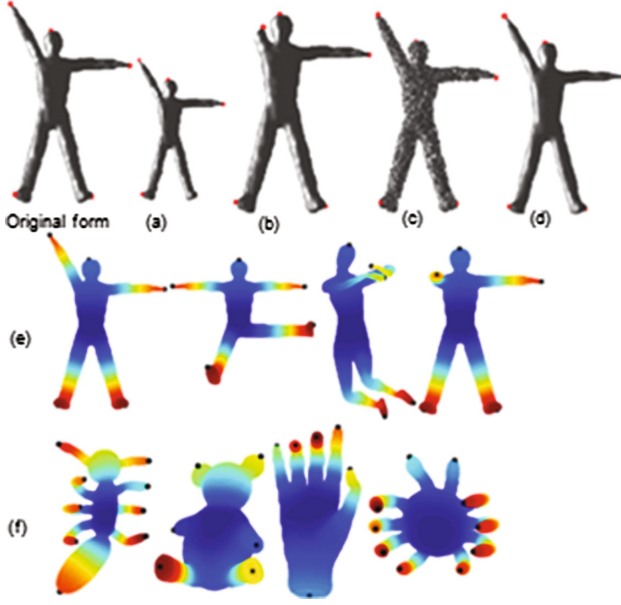


Fig. 1. Robustness for ADF detector to scale (a), occlusion (b), noise (c), smooth (d), non-rigid deformations (e) and its reliance on shape (f).

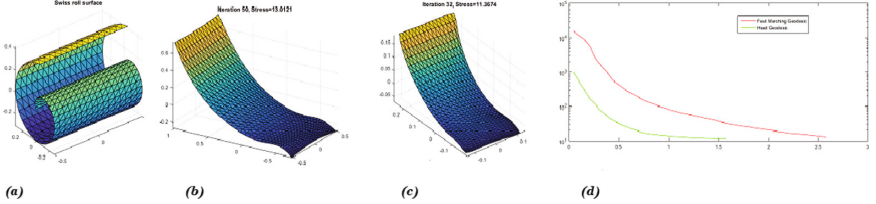


Fig. 2. Swiss roll surface (a), and its 3D canonical form using the classical MDS, (b) geodesic distance is computing by the fast marching algorithm and heat geodesic method(c). Their convergence speed is plotted is (d).

To preserve local features of the mesh, we do not treat all pairs of features in the same manner. However, we enforce a target weight between all connected vertices i and j in the same region. The value of $w_{i,j}$ is set according to the value of the commute-time distance $c_{i,j}$. Notes that this is the expected time taken by the random walk to travel from i to j in both directions. Therefore, if $c_{i,j}$ is small, then $\delta_{i,j}$ should also be small enough to minimize the stress function given in Eq. (1). The commute-time distance is defined as:

$$dc(i, j)^2 = \sum_{i=1} \frac{1}{\lambda_i} (h_i(x) - h_i(y))^2, \quad (8)$$

In order to assemble local canonical forms and create final canonical form of a given model, we need to add a constraint between different partitions. This constraint is formulated as the spatial relationship between the set of features points associated to the local patches.

As it is well known, the important properties of the distance are metric, smooth, locally isotropic, globally shape-aware, isometry invariant, insensitive to noise and small topology changes, parameter-free, and practical to compute on a discrete mesh. Thus, in order to satisfy our exigences, we made the choice of the biharmonic distance [9]. The latter distance, is based on the biharmonic differential operator, but applies different (inverse squared) weighting to the eigenvalues of the Laplace-Beltrami operator, which provides a nice trade-off between nearly geodesic distances for small distances and global shape-awareness for large distances. We used the discrete definition of the biharmonic distance based on the common cotangent formula discretization of the Laplace-Beltrami differential operator on meshes [18].

Given the discretization of the Laplacian, Lipman and all [9] defined the discrete Biharmonic distance as:

$$dB(x, y)^2 = \sum_{i=1} \frac{(h_i(x) - h_i(y))^2}{\lambda_i^2} \quad (9)$$

roughly speaking, we reformulate the final stress function to be minimised as:

$$S(X) = \sum_k \sum_{i,j \in P_k} \frac{c_{i,j}^2}{\delta_{i,j}^2} (\delta_{ij} - d_{i,j})^2 + \frac{1}{2} \sum_{i,j \in F} (ADF_i + ADF_j) (dB_{i,j} - d_{i,j})^2 \quad (10)$$

where k is the number of local patches and F is the set of feature points. $\delta_{i,j}$ is the local geodesic distances in each region P and $dB_{i,j}$ is the biharmonic distance between feature points.

We can ensure that our solution gives good results based on the comparison present in [10]. Thus, geodesic distance admits desirable local properties and gradually increases in the neighborhood of the source vertex (see Fig. 3). However, it is not globally shape-aware. In addition, the calculation of the biharmonic distances between extremities of the object is much faster than geodesic distances.

Figure 4 shows an overview of our algorithm (Fig. 4e) of a given 3D object (Fig. 4a). The scalar function ADF is used as an unsupervised detector of feature points (Fig. 4b and c). In Fig. 4c, we represent the overall geodesic distance computing from the local patches. For $n = 9501$ vertex, we use 17 million values in the dissimilarity matrix instead of n^2 , i.e. approximately a reduction of 80%. Examples of 3D canonical forms produced by our method are also given in Fig. 5 for different 3D models.

The computational complexity of the 3D canonical form methods depend on the time complexity of distances calculation (matrices δ and dB in Eq. (10) and the SMACOF algorithm (see Sect. 3).

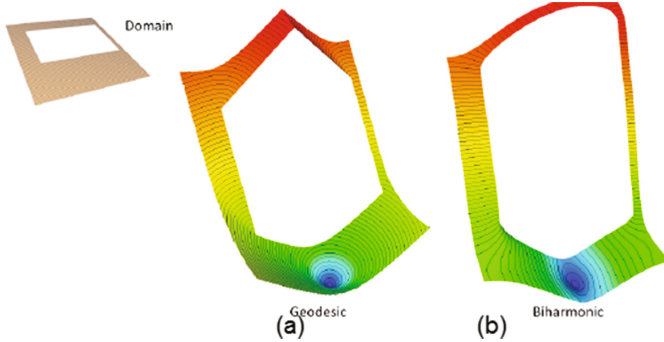


Fig. 3. Distances measured on an Euclidean domain (top-left). We visualize the distance field from a single source point to all other points as a height function over this Euclidean domain; (a) Geodesic distance, and (b) biharmonic distance. The geodesic distance possesses non-smooth curve of points and the isolines are not “shape-aware” far from the source. Biharmonic distance balances the “local” and “global” properties [10].

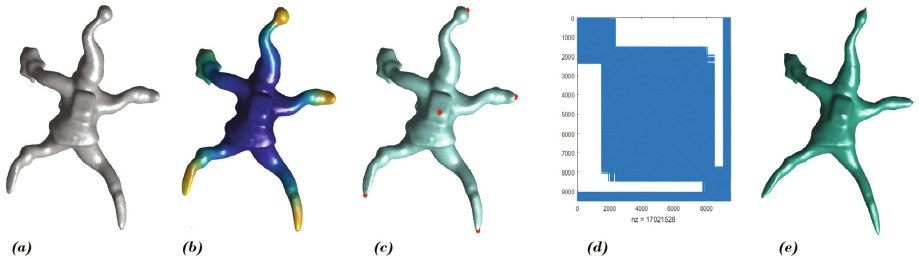


Fig. 4. Main steps of our procedure that employs the local features (c) extracted by the ADF scalar function defined on the mesh surface (b) to generate final 3D canonical form of the original model (a).

Geodesic based methods calculate the geodesic distance between all pairs of vertices using the fast marching algorithm which has a time complexity of $\mathcal{O}(N^2 \log N)$, where N is the total number of mesh vertices. Our method calculates geodesic distances between only points in the same local patch rather than all pairs. For a local patch on n vertices on average, with $n \ll N$, the time complexity is $\mathcal{O}(n^2 \log n)$. So, the total computational complexity of distance matrices has $\mathcal{O}(mn^2 \log n + m^2)$, where m is the number of local patches. Note that m is very small compared to N , (for instance, we have $m = 5$ and $N = 60000$, for the human body shape).

In addition, the SMACOF algorithm has a computational complexity of $\mathcal{O}(N^2)$, when the distance between all pairs of points is used. Our Algorithm lowers this complexity to $\mathcal{O}(mn^2)$.



Fig. 5. 3D Canonical forms for a selection of the SHREC'15 dataset produced by our method.

5 Experiments

5.1 Canonical Forms

We base the evaluation of the performance of the proposed parameter-free method for 3D canonical forms computation on the SHREC'15 Track [17]. The later is a new benchmark for testing algorithms at creating canonical forms for use in non-rigid 3D shape retrieval.

The dataset contains models from both the SHREC11 non-rigid benchmark [19] and the SHREC14 non-rigid humans benchmark [20]. The total number of meshes in the dataset is 100 meshes, split into 10 different shape classes. Each shape class contains a mesh in 10 different non-rigid poses.

Authors 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. They proposed two measures:

- The compactness measure is calculated as $\frac{V^2}{A^3}$, where V is the mesh volume and A is the mesh surface area. The difference in the compactness measure between the original mesh and the canonical form quantify the amount of distortion of the original shape.
- The Hausdorff distance is computed using iterative closest point matching to align each pair of shapes of the same class. This measure quantify the consistency of the canonical forms across models of the same shape.

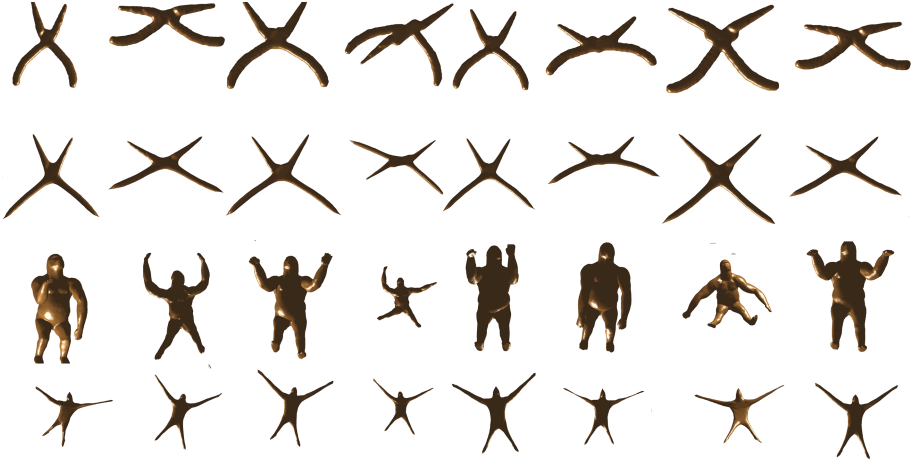


Fig. 6. Example of canonical forms of each mesh.

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

Class	Mean Error	Standard deviation
Santa	1.994	0.633
Horse	4.344	1.360
Dog1	1.907	0.309
Bird2	0.586	0.672
Laptop	0.336	0.741
Female1	5.726	1.721
Female2	5.281	1.390
Male1	4.551	1.076
Child body	3.963	1.441
Male2	4.246	1.445
Whole	3.3246	2.102

Several canonical forms of two objects with non-rigid deformations are shown in Fig. 6. Our method successfully producing canonical forms of each shape by eliminating non-rigid deformations and stretched out their extremities. The obtained results aim to standardize its pose.

Table 1 shows the compactness measure of our method for each class in the dataset at preserving the shape of the original model. Our method performs better at preserving compactness of all the dataset. This highlights the advantage of using local features instead of global features.

Table 2. Hausdorff distances between canonical forms of the same class.




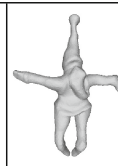


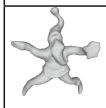
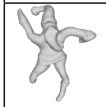
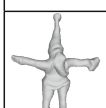
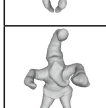
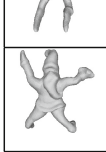
					
	0	0.181	0.182	0.181	0.165
	0.181	0	0.181	0.181	0.1657
	0.182	0.181	0	0.1658	0.1657
	0.181	0.181	0.1658	0	0.1658
	0.165	0.1657	0.1657	0.1658	0

Table 3. Error in compactness between the canonical forms and the original meshes for each method. The values have been multiplied by 10^4 .

	Classical MDS	Fast MDS	Least-squares MDS	Non-metric MDS	Skeletons	Euclidean random	Euclidean normalised	1-Form MDS	Our method
Mean Error	8.44	6.60	4.77	5.17	0.71	1.86	2.97	4.89	3.32
Standard deviation	4.420	3.384	3.595	3.754	0.907	1.362	1.231	3.712	2.102

Furthermore, we used the Meshlab [21] to compute the Hausdorff distance between a few of models of the same shape class of the SHREC'15 dataset. Table 2 shows small distances between the associated canonical forms, despite the presence of non-rigid deformations. These results promote the use of canonical forms in the non-rigid object recognition task.

In addition, we compare our results against those submitted to the SHREC'15 canonical forms benchmark [17]. Table 3 shows that our local method outperforms geodesic based methods and cause less shape distortion. However, our method is competitive with methods based on Euclidean distance. Yet, we only

use local features at the ends of the shape members. In addition, the parameters of our method are fully automatic and free.

6 Conclusion

In this paper, we presented a novel method to construct canonical forms for 3D models based on the classical MDS method. Our contribution is to divide the problem of 3D canonical form embedding into sub-problems. To solve the resulting optimisation sub-problems, we added a spatial constraint between local features. We took advantage of the good properties of the Biharmonic distance to add relationship between sub-problems. In addition, we proposed a dynamic setting of the weight values in the stress function, according to the values of the dissimilarity matrix and the commute-time weight.

Experiments demonstrated the efficiency of our algorithm to construct an invariant 3D canonical form for a 3D surface. Our method preserve local and global features between original surfaces and MDS embedded surfaces and is also able to achieve the same bending invariant pose as the previous state-of-the-art. Yet, it involves far less shape distortion than other geodesic based methods.

As future works, we propose to test the performance of our method in the retrieval task on a recent canonical forms benchmark.

References

1. Cox, M.A., Cox, T.F.: *Multidimensional Scaling*. Chapman and Hall, London/ New York (1994)
2. Borg, I., Groenen, P.: *Modern Multidimensional Scaling: Theory and Applications*. Springer, Heidelberg (1997)
3. Faloutsos, C., Lin, K.D.: A fast algorithm for indexing, data mining and visualisation of traditional and multimedia datasets. In: *Proceedings of ACM SIGMOD*, pp. 163–174 (1995)
4. Katz, S., Leifman, G., Tal, A.: Mesh segmentation using feature point and core extraction. *Vis. Comput.* **21**, 8–10 (2005)
5. Elad, A., Kimmel, R.: On bending invariant signatures for surface. *IEEE Trans. Pattern Anal. Mach. Intell.* **25**(10), 1285–1295 (2003)
6. Lian, Z., Godil, A., Xiao, J.: Feature-preserved 3d canonical form. *Int. J. Comput. Vis.* **102**, 221–238 (2013)
7. Wang, X.-L., Zha, H.: Contour canonical form: an efficient intrinsic embedding approach to matching non-rigid 3D objects. In: *Proceedings of the 2nd ACM International Conference on Multimedia Retrieval*, Article No. 31 (2012)
8. Pickup, D., Sun, X., Rosin, P.L., Martin, R.R.: Euclidean-distance-based canonical forms for non-rigid 3D shape retrieval. *Pattern Recognit.* **48**(8), 2500–2512 (2015)
9. Lipman, Y., Rustamov, R.M., Funkhouser, T.A.: Biharmonic distance. *ACM Trans. Graph. (TOG)* **29**(3), 27 (2010)
10. Geřbal, K., Bærentzen, J.A., Aanæs, H., Larsen, R.: Shape analysis using the auto diffusion function. *Comput. Graph. Forum* **28**(5), 1405–1413 (2009)
11. Mohamed, H.H., Belaid, S.: Algorithm BOSS (Bag-of-Salient local Spectrums) for non-rigid and partial 3D object retrieval. *Neurocomputing* **168**, 790–798 (2015)

12. Papadimitriou, C.H.: An algorithm for shortest-path motion in three dimensions. *IPL* **20**, 259–263 (1985)
13. Crane, K., Weischedel, C., Wardetzky, M.: Geodesics in heat: a new approach to computing distance based on heat flow. *ACM Trans. Graph.* **32**(5), Article No. 152 (2013)
14. Bronstein, A.M., Bronstein, M.M., Kimmel, R.: *Numerical Geometry of Non-Rigid Shapes*. Monographs in Computer Science. Springer, Heidelberg (2009)
15. Kimmel, R., Sethian, J.A.: Computing geodesic paths on manifolds. *Proc. Natl. Acad. Sci.* **95**, 8431–8435 (1998)
16. Surazhsky, V., Surazhsky, T., Kirsanov, D., Gortler, S.J., Hoppe, H.: Fast exact and approximate geodesics on meshes. *ACM Trans. Graph.* **24**(3), 553–560 (2005)
17. Pickup, D., et al.: SHREC 15 track: canonical forms for non-rigid 3D shape retrieval (2015)
18. Grinspun, E., Gingold, Y., Reisman, J., Zorin, D.: Computing discrete shape operators on general meshes. *Eurograph. Comput. Graph. Forum* **25**(3), 547–556 (2006)
19. Lian, Z., Godil, A., Bustos, B., Daoudi, M., Hermans, J., Kawamura, S., Kurita, Y., Lavoue, G., Nguyen, H.V., Ohbuchi, R., Ohkita, Y., Ohishi, Y., Porikli, F., Reuter, M., Sipiran, I., Smeets, D., Suetens, P., Tabia, H., Vandermeulen, D.: SHREC 11 track: shape retrieval on non-rigid 3D watertight meshes. In: *Proceedings of the 4th Eurographics Conference on 3D Object Retrieval*, pp. 79–88 (2011)
20. Cheng, Z., Lian, Z., Aono, M., Hamza, A.B., 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.: Shape retrieval of non-rigid 3D human models. In: *Proceedings of the 7th Eurographics Workshop on 3D Object Retrieval*, pp. 101–110 (2014)
21. Meshlabv1.3.0 (2011). <http://meshlab.sourceforge.net>