Bending Invariant Representations for Surfaces

ASI ELAD (ELBAZ) RON KIMMEL
Department of Computer Science
Technion–Israel Institute of Technology
Technion City, Haifa 32000, Israel
asi.elad@excite.com

Abstract

Isometric surfaces share the same geometric structure also known as the 'first fundamental form'. For example, all possible bending of a given surface, that include all length preserving deformations without tearing or stretching the surface, are considered to be isometric. We present a method to construct a bending invariant *canonical form for such surfaces. This invariant representation is an embedding of the intrinsic geodesic structure of the surface in a finite dimensional Euclidean space, in which geodesic distances are approximated by Euclidean ones. The canonical representation is constructed by first measuring the inter geodesic distances between points on the surfaces. Next, multi-dimensional scaling (MDS) techniques are applied to extract a finite dimensional flat space in which geodesic distances are represented as Euclidean ones. The geodesic distances are measured by the efficient 'fast marching on triangulated domains' numerical algorithm. Applying this transform to various objects with similar geodesic structures (similar first fundamental form) maps isometric objects into similar canonical forms. We show a simple surface classification method based on the bending invariant canonical form.*

1. Introduction

The problem of finding a full or partial match between surfaces attracted the attention of computer vision and pattern recognition researchers during the past decade. Most of the existing techniques, like Potmeisl [1], address rigid object matching by heuristic algorithms that search for the transformation that maximizes shape similarities while registrating the two objects. Besl [2, 3] proposes metrics for measuring matches between curves and surfaces. Faugeras [4], and Faugeras and Hebert [5] used *quaternions* to transform the 3D rotation problem into a four-dimensional minimum eigenvalue problem, while the translation is computed using a standard least-squares technique. Lavallee and Szeliski [6] solve the 2D/3D matching problem by a least-squares minimization of the 'energy' required to align the projection lines of the camera contours tangent to the object. Barequet and Sharir [7] associate a *footprint* for each surface point in order to extract separately the rotation and translation components of the desired rigid transformation. A survey of these techniques can be found in [7] and [8].

In this paper we propose a different solution to a different surface matching problem. While previous methods dealt mostly with rigid transformations or rely on key points and local or semi-differential invariant measures, here we address the bending invariant problem by a transformation that takes isometric surfaces to similar manifolds in a finite dimensional flat Euclidean space. In the Euclidean space, these manifolds can be compared using existing matching techniques. Our method is based on two numerical procedures. The first, is the *fast marching on triangulated domains* [9], that efficiently calculates geodesic distances on triangulated curved surfaces. The second, is the Multi-Dimensional Scaling (MDS) [10, 11, 12], that uncovers the geometric structure of a set of data items from a (dis)similarity information among them. The outline of this paper is as follows: Section 2 is brief review of the fast marching on triangulated domains algorithm. Section 3 presents the basic concepts of MDS and reviews various MDS methods like the *Classical*, *Least-Squares*, and the *Fast MDS*, that were tested and compared. Section 4 explains how these methods can be used to solve the matching problem for isometric objects. Experimental results are presented in Section 5.

2. Fast Marching on Triangulated Domains

The proposed technique matches between isometric surfaces, or in other words, surfaces for which the geodesic distances between corresponding surface points are the same. Therefore, we base our method on the intrinsic geodesic distances between surface points. We first compute the geodesic distances between pairs of points on the surface. The *fast marching method*, introduced by Sethian [13] is a

numerically consistent distance computation approach that works on rectangular grids, see also Tsitsiklis [14] Eikonal solver on rectangular grids for a related approach.It was extended to triangulated domains by Kimmel and Sethian in [9]. The basic idea is an efficient numerical approach that solves the Eikonal equation $|\nabla u| = 1$, where at the source point s the distance is zero, namely $u(s) = 0$. The solution u is a distance function and its numerical approximation is computed by a monotone update scheme that is proven to converge to the 'viscosity' smooth solution. The idea is to iteratively construct the distance function by patching together small plans supported by neighboring grid points with gradient magnitude that equals one. The distance function is constructed by starting from the sources point, s, and propagating outwards. Applying the method to triangulated domains requires a careful analysis of the update of one vertex in a triangle, while the u values at the other vertices are given. For further details we refer to [9].

The fast marching on triangulated domains method can compute the geodesic distance between one vertex and the rest of the n surface vertices in $O(n)$ operations. Repeating this computation for each vertex, we compute a geodesic distance matrix D in $O(n^2)$ operations. Each ij entry of D represents the square geodesic distance between the vertex i and the vertex j, that is $\delta_{ij} = \mathrm{GeodesicDistance}(\mathrm{Vertex}_i, \mathrm{Vertex}_j)$, where $[D]_{ij} = \delta_{ij}^2$. Practically, in order to reduce the computational effort, we select a sub-set of the surface vertices using a variation of the technique proposed in [15]. Thereby, given a triangulated surface, we apply the fast marching procedure for each vertex in a subset of the vertices as a source point, and obtain the geodesic distance matrix, D.

3. MDS

Multi-Dimensional Scaling (MDS) is a family of methods that maps measurements of similarity or dissimilarity among pairs of feature items, into distances between feature points with given coordinates in a small-dimensional Euclidean space. The graphical display of the (di)similarity measurements provided by MDS enables the analyst to literally 'look' at the data and explore its geometric structure visually. Most metrical MDS methods expect a set of n items and their pairwise (dis)similarities, and the desirable dimensionality - m of the Euclidean embedding space.

The algorithm maps each item to a point x_i in an m dimensional Euclidean space $\mathbf{R}^m$ by minimization of, for example, the stress function

$$Stress = \frac{\sum_{i<j} w_{ij}(\delta_{ij} - d_{ij})^2}{\sum_{i<j} \delta_{ij}^2}, \qquad (1)$$

where δ_{ij} is the input dissimilarity measure between item i and j, d_{ij} is the Euclidean distance between these items in the m-dimensional Euclidean space, and w_{ij} are some weighting coefficients. Here, we use MDS in the following way. As proximity values we use the squared geodesic distance calculated by the *fast marching on triangulated domains*.

Let us start with a simple example in which we set apriori $m = 3$. Given the dissimilarities matrix, an MDS method yields coordinates, X, in $\mathbf{R}^3$, for which the Euclidean distances between the points in $\mathbf{R}^3$ would be similar to the geodesic distances between the corresponding surface vertices. The Euclidean distance between each pair of points in X would be as close as possible to the geodesic distance between the corresponding points extracted from the data, that in our case represent the geodesic distance on the surface. Given the connectivity of the vertices as triangles that represent the curved surface, we can connect the corresponding points after the MDS flattening and obtain a manifold that we refer to as *bending invariant canonical form* of the surface.

The selection of $\mathbf{R}^3$ was arbitrary in this example. In order to select a 'proper' dimension, m, we define the 'effective' dimensionality of the problem to be m, if for a higher dimension, $m+1$ the norm of the error is improved by less than $p\%$. Where for example we use $p \leq 5\%$ in our experiments. If we want to graphically view an invariant 'canonical form,' the number of dimensions should be less than four.

3.1. Classical MDS

Classical scaling was originated in the 1930's when Young et al. showed that given a matrix of distances between points in an Euclidean space, one can extract coordinates such that distances are preserved, see e.g. [11]. Let the coordinates of n points in a k dimensional Euclidean space $\mathbf{R}^k$ be given by x_r, $(r = 1, ..., n)$, where $x_r = [x_{r1}, x_{r2}, ..., x_{rk}]^T$. The Euclidean distance between the r-th and the s-th points is given by $d_{rs}^2 = [x_r - x_s]^T[x_r - x_s]$. Let the inner product matrix be B, where the rs element is given by $[B]_{rs} = b_{rs} = x_r^T x_s$. Given the squared distances matrix D, the inner product matrix is given by (see [10]) $B = -\frac{1}{2}JDJ$, where $J = I - \frac{1}{n}\mathbf{1}\mathbf{1}^T$ and $\mathbf{1}_{1\times n} = [1, 1, ..., 1]$. We also have that $B = XX^T$, where $X = [x_1, ...x_n]^T$ is the $n \times k$ matrix of the coordinates. The inner product matrix B is symmetric, positive semi-definite and of rank k. Therefore, B has k non-negative eigenvalues and $n - k$ zero eigenvalues. The matrix B can be expressed in terms of its spectral decomposition, $B = V\Lambda V^T$ where $\Lambda_{n\times n} = \mathrm{diag}(\lambda_1, \lambda_2, ..., \lambda_k, 0...., 0)$. For convenience, the eigenvalues of B are ordered such that $\lambda_1 \geq \lambda_2 \geq ....\lambda_k \geq 0$. Hence, the required coordinates are given by using the non-zeros sub-matrix $\Lambda_{k\times k}$,

$$X = V\Lambda_{k\times k}^{\frac{1}{2}}.$$

The classical scaling is considered to be an efficient al-

Obj/Dim	$m=1$	$m=2$	$m=3$	$m=4$	$m=5$
Torus	73.5	88.7	95.1	100	100
Human	53.45	87.3	100	100	100
Rabbit	81.1	92.9	98	100	100
Elephant	57.54	74.51	91.1	96.5	100

Table 1: Classical MDS - energy percentage of the first m eigenvalues.

Obj/Dim	$m=1$	$m=2$	$m=3$	$m=4$	$m=5$
Torus	12.96	0.0714	0.0098	0.0098	0.0098
Human	3.695	0.0341	0.0051	0.0051	0.0051
Rabbit	1.99	0.044	0.0029	0.0021	0.0021
Elephant	3.42	0.113	0.0043	0.0036	0.0036

Table 2: LS MDS: The stress (1) as a function of the number of dimensions m.

gebraic approach to solve the MDS problems. It can be calculated in $O(n^2)$, where n is the number of feature points in the given model. This is due to the fact that there is a need to find only the first m eigenvalues and the corresponding eigenvectors, which can be computed by variations of the 'power method', see for example [12]. Instead of the stress function (1), classical MDS approach minimizes the measure given by

$$E = \|V \cdot (\Lambda - \widetilde{\Lambda}) \cdot V^T\|, \tag{2}$$

where

$$\begin{aligned} \widetilde{\Lambda} &= \mathrm{diag}(\lambda_1, \lambda_2, ..., \lambda_m, 0...0). \\ \Lambda &= \mathrm{diag}(\lambda_1, \lambda_2, ..., \lambda_m, .., \lambda_k, 0...0), \end{aligned}$$

and $m \leq k$.

An empirical analysis of the 'effective' dimensionality of the problem is given in Table 1. It shows that selecting three to five dimensions capture $95\% - 100\%$ of the error norm for typical surfaces in $\mathbf{R}^3$. As can be easily extracted form (2), the norm of the error is the sum of the last $(k-m)$ eigenvalues divided by the total sum of the eigenvalues. As mentioned, limiting the dimensions to $m \leq 3$, enables us to graphically view the 'canonical' manifold. Usually, when analyzing smooth surfaces, most of the geometric structure is captured by the first three eigenvalues.

3.2. Least Squares MDS

The Least Squares technique is a standard optimization approach to solve the minimization problem of the cost defined by the stress function (1). The problem is that there is no simple way to form a close expression for the first derivative of this non-linear functional. An simple yet powerful minimization strategy is the principle of minimizing a function by iterative majorization. This method is applied in the SAMCOF (Scaling by Maximizing a Convex Function) algorithm for minimizing stress [11]. The idea is to bound the stress function $S(X)$ iteratively by a simple function $\hat{S}(X, Z)$, where Z is a possible solution, and $\hat{S}(X, Z) \geq S(X)$ for $X \neq Z$, and $\hat{S}(Z, Z) = S(Z)$. Let us briefly review this idea. For further details see [11].

Minimizing the stress (1) is equal to minimizing the following functional $S(x) = \sum_{i<j} w_{ij}(\delta_{ij} - d_{ij})^2$, or $S(x) = \eta_\delta^2 + \eta(X)^2 - 2\rho(X)$. Applying *Cauchy-Schwartz* inequality and basic algebraic operations, we have that the stress function can be bounded by the following quadratic function in X,

$$S \leq \eta_\delta^2 + \mathrm{tr} X^T V X - 2\mathrm{tr} X^T B(Z) Z = \tau(X, Z).$$

Where the matrices V, Z, and $B(Z)$ depend on $\{\delta_{ij}\}$, the weight matrix W and the current X_i solution. The minimum of $\tau(X, Z)$ can be achieved by setting the derivative of $\tau(X, Z)$ to zero, and the required solution is given by

$$X_i = V^+ B(Z) Z. \tag{3}$$

Where the matrix V^+ is the Moore-Penrose inverse of V. It can be shown that if all weights $w_{ij} = 1$, then the update simplifies to

$$X_i = n^{-1} B(Z) Z. \tag{4}$$

The SAMCOF algorithm for MDS can be summarized by the following steps,

1. Set $Z = X_0$ and $i = 0$, where X_0 is a (non)random initial configuration.
2. Compute the stress function $S(X_0)$.
3. Set $i = i + 1$.
4. Compute the next solution X_i by (3) or (4).
5. If $S(X_i) - S(X_{i-1}) < \varepsilon$ then stop.
6. Set $Z = X_i$ and go to step 3.

Considering more than three dimensions, as illustrated in Table 2, decrease the stress (1) by less than 1% in our examples. In our examples the LS MDS required less than a hundred iterations to converge. Hence, the complexity is of $O(n^2 \cdot \mathrm{NumOfIterations})$.
In our case where the distances matrix consists of the geodesic distances, the Least Square method that minimizes the stress function, better captures the structure of the geometric data in some cases compared to the classical scaling results.

3.3. Fast MDS

The Fast MDS is a recent heuristic technique proposed by Lin and Faloutsos [16]. This method is computationally efficient, $O(nm)$ where m is the target dimensions, that can be considered to be $O(n)$ in our case. Yet, it does not minimize any global measure, but only attempts to approximate it. This technique works recursively by generating a new

dimension at each step, providing m-dimensional coordinates after applying the recursion m times. The basic idea is to project the vertices on a selected 'line'. First, the algorithm selects two vertices O_a and O_b, that should be as far as possible from one other. Next, all other vertices are projected on that line using the *cosine law* $x_i = \frac{d_{ai}^2 + d_{ab}^2 - d_{bi}^2}{2d_{ab}}$, see Figure 1.

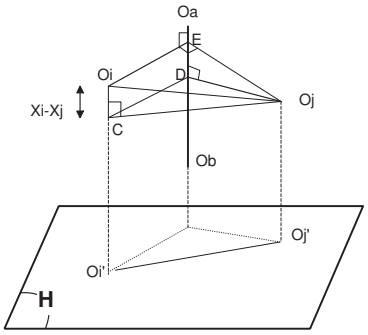

Figure 1: Projection on the hyper-plane H.

The next step is to project all items to an $(n-1)$ hyperplane H that is perpendicular to the line (O_a, O_b) and regenerate a new distance matrix according to $d_{i'j'}^2 = d_{ij}^2 - (x_i - x_j)^2$. This step should be repeated m times, where at each step the calculated x_i, $i = (1, 2...., n)$ are the new added dimension coordinates.

The m-dimensional coordinates can be calculated in $O(mn)$, provided that at each step we use a linear heuristic algorithm to choose the pivot feature items.

We tested the three techniques, Classical, LS and Fast MDS, and obtained the results in Figures 2, 3, 4 and 5, for $m = 3$. As expected, the heuristic fast method yields the least accurate results yet was the fastest, while the LS is the most accurate technique yet the slowest. Table 3 summarizes the stress (1) for increasing dimensions of the three MDS techniques. The LS method reaches the minimal stress, which is not surprising since the classical MDS minimizes another measure as described above. However, we can see that the stress decreases for all three methods as the number of dimensions m gets larger.

4. Matching Surfaces

Equipped with the fast marching on triangulated domains, and the MDS, we can solve the isometric surface matching problem. For example, given objects in 3D, some of which are isometric, we would like to measure their isometric-(dis)similarity and thereby classify them.

In the first step we compute the geodesic distance matrix for each object using the fast marching on triangulated domains. As stated above, given an object with n vertices we

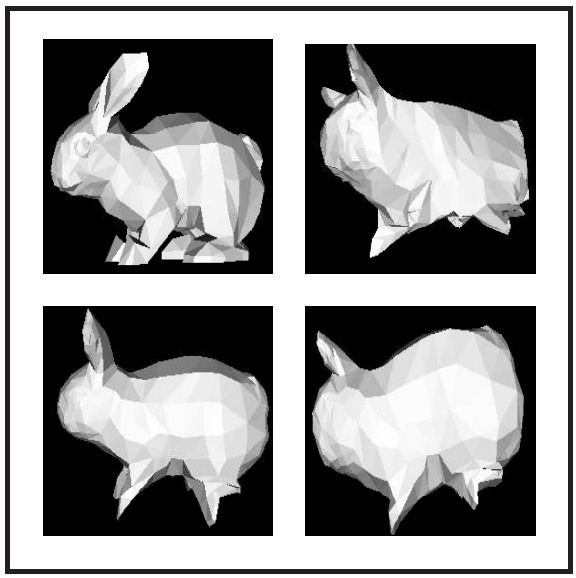

Figure 2: Rabbit. Top left: Original surface. Top right: Fast MDS result. Bottom left: LS MDS result. Bottom right: Classical MDS result.

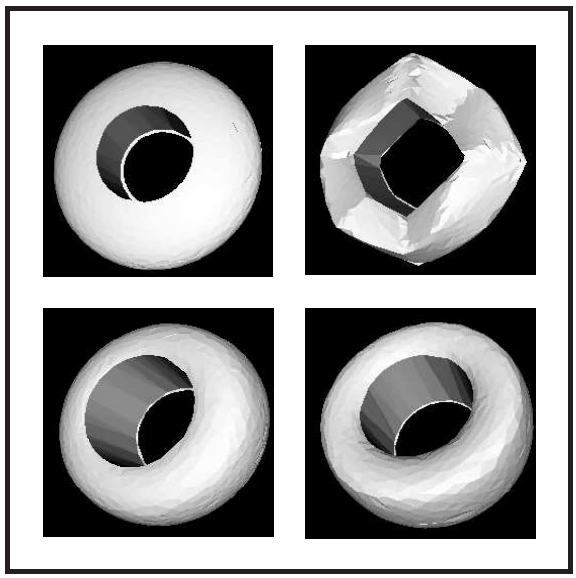

Figure 3: Torus. Top left: Original surface. Top right: Fast MDS result. Bottom left: LS MDS result. Bottom right: Classical MDS result.

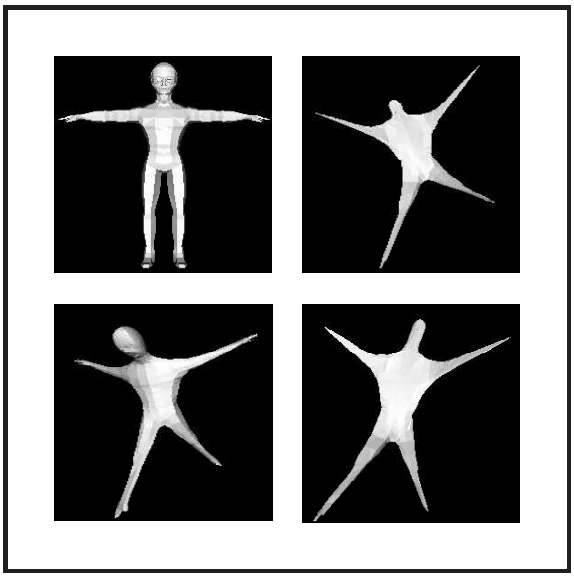

Figure 4: Human body. Top left: Original surface. Top right: Fast MDS result. Bottom left: LS MDS result. Bottom right: Classical MDS MDS result.

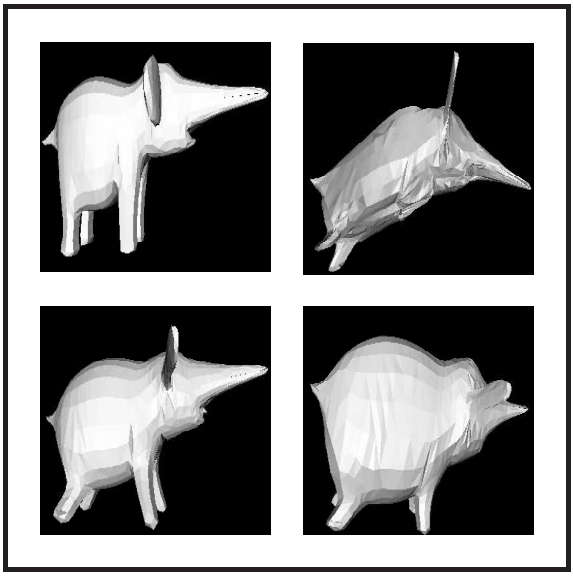

Figure 5: Bending invariant canonical representation for an elephant. Top left: Original surface. Top right: Fast MDS result. Bottom left: LS MDS result. Bottom right: Classical MDS result.

Obj/Dim	$m = 1$	$m = 2$	$m = 3$	$m = 4$	$m = 5$
Torus					
Classical	68.59	0.0801	0.0122	0.0122	0.0122
LS	12.96	0.0714	0.0098	0.0098	0.0098
Fast	39.8	0.1151	0.0136	0.0104	0.0104
Human					
Classical	12.70	0.0643	0.0176	0.0176	0.0176
LS	3.695	0.0341	0.0051	0.0051	0.0051
Fast	9.626	0.04392	0.0201	0.0161	0.0161
Rabbit					
Classical	15.26	0.2621	0.0097	0.0077	0.0077
LS	1.356	0.081	0.0033	0.0031	0.0031
Fast	5.475	0.206	0.0205	0.0119	0.0081
Elephant					
Classical	10.68	0.2954	0.0114	0.0073	0.0076
LS	3.42	0.113	0.0043	0.0036	0.0036
Fast	13.28	0.3051	0.0381	0.0122	0.0077

Table 3: The stress (1) for LS, Classical, and Fast MDS as a function of m, the embedding Euclidean dimension.

select a subset of $\widetilde{n} \leq n$ vertices, and calculate the geodesic distances between each pair in this set using the original surface. The sub-sampling technique is an iterative process where in each iteration the farthest (in the geodesic sense) vertex from the already selected vertices is selected. The process starts by selecting the first vertex randomly, and terminates when the sub-set of selected vertices reached a pre-defined number.

An allowable mapping of the surface S^1 onto the surface S^2 is said to be *isometric* or *length preserving* if the length of any arc on S^2 is the same as that of its inverse image on S^1, [17]. In order to extract a practical algorithm from the above definition we take the following approximation steps. First, we sample the surface and consider its triangulated version as an approximate geometric representation of the continuous one. Next, for computational efficiency of the MDS-flattening step we consider only a subset of the given vertices. Both these approximation steps were verified empirically to introduce acceptable deformations in the results.

The flattening step involves setting the required dimension, m, and application of the MDS techniques on the geodesic distances matrices. It extracts the coordinates of the objects in an invariant 'canonical' form. We can also handle similarity (scaling) transformations by normalizing the canonical manifold: After the flattening, the canonical manifolds are uniformly scaled to the same bounding box, and centralized (done automatically in the classical MDS) and oriented using the second (the eigenvalues in the classical MDS) and the third order moments.

Next, we have to compare between the 'canonical' man-

ifolds that are given by their coordinates in m dimensions. Again, we can construct a distance matrix between the given manifolds based on some unique measure, like the Hausdorff distance. A simple matching measure was proposed by Elad and Tal in [18]. They consider only the first m ($m < 15$) moments of each surface (the invariant manifold in our case) and calculate the Euclidean distances between each pair. This operation yields a moments-distance matrix, D, where $[D]_{ij} = \mathrm{distance}(\mathrm{Object}_i, \mathrm{Object}_j)$.

Applying MDS process on this new matrix yields points in a Euclidean space, where each point represents one surface. This way, isometric surfaces are clustered together while non-isometric surface are well separated in this Euclidean space, as will be graphically illustrated in the next section.

5. Experimental Results

We applied the proposed algorithm to the objects shown in Figure 6. The input surfaces include six shapes of a human body, a hand, a hat, a paper, a dog, spider, and bending versions of these shapes. For the flattening results shown in Figure 8, we used the Least Squares MDS. For presentation purposes, justified by the error norm, we selected the Euclidean embedding space to be of three dimensions. One can see that isometric surfaces are mapped to similar 'canonical' manifolds. The results of applying the moments based clustering step described in the previous section, is shown in Figure 9. As a reference, Figure 7 illustrates the results of applying the moment based MDS clustering procedure to the original surfaces.

Isometric objects are closer to one another compared to the flattening result of the moment distances of the original surfaces. The MDS based clustering and classification steps are only an example application for using the bending invariant canonical representations of the surfaces. Other matching techniques that work on the flattened invariant canonical form are possible.

6. Conclusions

An efficient method for computing bending invariant canonical representations of surfaces was presented. The method is based on the fast marching on triangulated domains algorithm followed by a multi-dimensional scaling (MDS) flattening technique. The 'canonical' form is computed by applying an MDS procedure on the geodesic distances matrix computed by the fast marching algorithm. Three different MDS techniques were tested, the Classical, the Least Squares and the Fast MDS, with the LS being the most accurate yet the slowest method. Our approach followed by a simple clustering algorithm was shown to be useful for 3D non-rigid isometric objects matching and classification.

References

[1] M. Potmesil. Generation of 3D surface descriptions from images of pattern-illuminated objects. *Proc. IEEE conf. Pattern Recognition and Image Processing*, pages 553–559, 1979.

[2] P. J. Besl. Geometric modeling and computer vision. *Proc. IEEE*, 76:936–958, 1988.

[3] P. J. Besl. The free form matching problem. *Machine Vision for Three-Dimensional Scene, H. Freeman, ed. New York Aacademic*, 1990.

[4] O. D. Faugeras. New steps towards a flexible 3D vision system for robotics. *Proc. IEEE Seventh Int'l Conf. Pattern Recognition*, 2:796–805, July 30 - Aug. 2 1984.

[5] O. D. Faugeras and M. Hebert. A 3D recognition and positioning algorithm using geometrical matching between primitive surfaces. *Proc. Seventh Int'l Joint Conf. Artificial Intelegence*, pages 996–1002, Aug 1983.

[6] S. Lavalee and R. Szeliski. Recovering the position and orientation of free-form objects from image contours using 3D distance map. *IEEE Trans. Pattern Analysis ans Machine Intelegence*, 17(4):378–390, 1995.

[7] G. Barequet and M. Sharir. Recovering the position and orientation of free-form objects from image contours using 3D distance map. *IEEE Trans. Pattern Analysis ans Machine Intelegence*, 19(9):929–948, 1997.

[8] P. J. Besl and R. C. Jain. Three-dimensional object recognition. *ACM Computing Surveys*, 17(1):75–154, 1985.

[9] R. Kimmel and J.A Sethian. Computing geodesic on manifolds. *Proc. of National Academy of Science*, 95:8431–8435, 1998.

[10] M.A.A. Cox and T.F. Cox. *Multidimensional Scaling*. Chapman and Hall, 1994.

[11] I. Borg and P. Groenen. *Modern Multidimensional Scaling - Theory and Applications*. Springer, 1997.

[12] J. B. Kruskal and M. Wish. *Multidimensional Scaling*. Sage, 1978.

[13] J.A Sethian. A review of the theory, algorithms, and applications of level set method for propagating surfaces. *Acta Numerica, Cambridge University Press*, 1996.

[14] J. N. Tsitsiklis. Efficient algorithms for globally optimal trajectories. *IEEE Trans. on Automatic Control*, 40:1528–1538, 1995.

[15] S. Melax. A simple, fast and effective polygon reduction algorithm. *Game Developer Journal*, November 1998.

[16] C. Faloutsos and K. D. Lin. A fast algorithm for indexing, data-mining and visualisation of traditional and mutimedia datasets. *ACM SIGMOD San Jose CA*, pages 163–174, May 1995.

[17] E. Kreyszig. *Differential Geometry*. Dover, 1991.

[18] A. Tal and M. Elad. Similarity between 3D objects - an iterative and interactive approach. *Submmited to IEEE Trans on Neural Networks*, 2000.

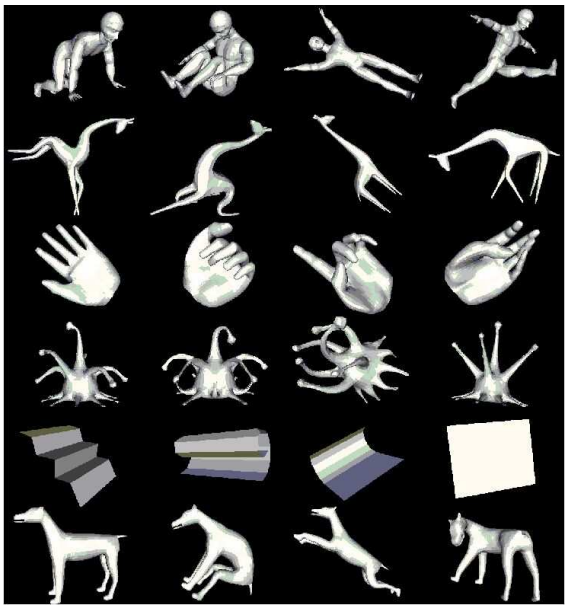

Figure 6: Six input surfaces of a human body, a hand, a hat, a paper, a dinosaur and a spider, and a few bending versions of these surfaces.

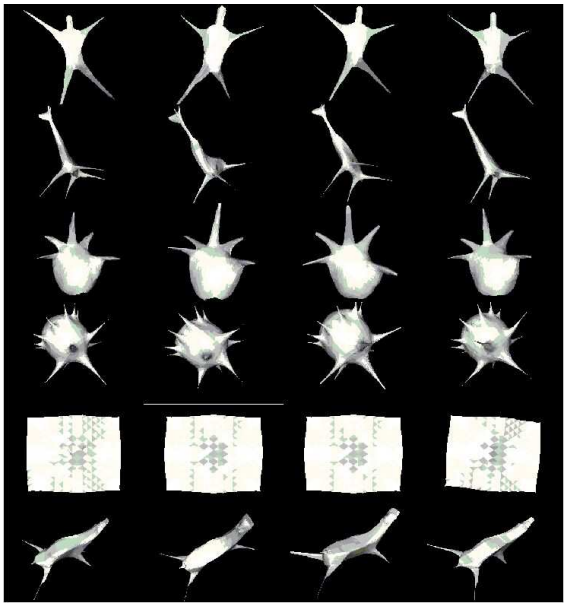

Figure 8: Output - The invariant canonical forms based on LS MDS.

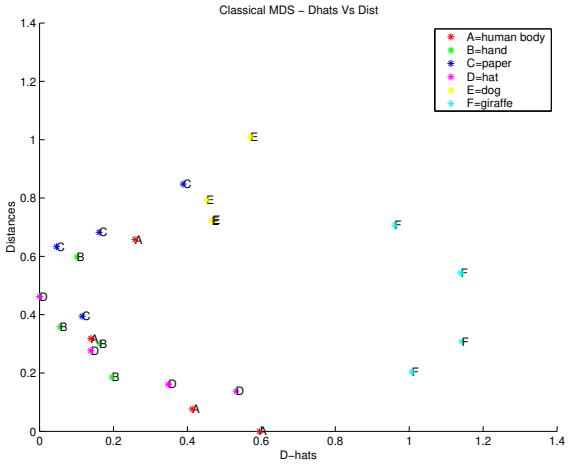

Figure 7: Applying the moments based clustering to the original manifolds.

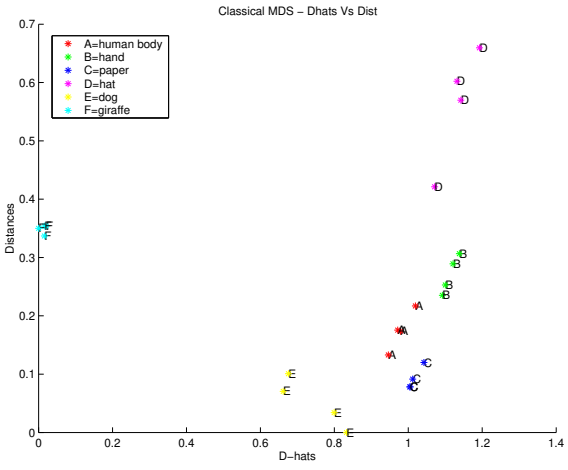

Figure 9: Applying the moments based clustering to all canonical manifolds for $m = 3$.