On Bending Invariant Signatures for Surfaces

Asi Elad (Elbaz) and Ron Kimmel, *Senior Member*, *IEEE*

Abstract—Isometric surfaces share the same geometric structure, also known as the "first fundamental form." For example, all possible bendings of a given surface that includes all length preserving deformations without tearing or stretching the surface are considered to be isometric. We present a method to construct a *bending invariant signature* for such surfaces. This invariant representation is an embedding of the geometric structure of the surface in a small dimensional Euclidean space in which geodesic distances are approximated by Euclidean ones. The bending invariant representation is constructed by first measuring the intergeodesic distances between uniformly distributed points on the surface. Next, a multidimensional scaling (MDS) technique is applied to extract coordinates in a finite dimensional Euclidean space in which geodesic distances are replaced by Euclidean ones. Applying this transform to various surfaces with similar geodesic structures (first fundamental form) maps them into similar signature surfaces. We thereby translate the problem of matching nonrigid objects in various postures into a simpler problem of matching rigid objects. As an example, we show a simple surface classification method that uses our bending invariant signatures.

Index Terms—MDS (Multi-Dimensional Scaling), FMTD (Fast Marching Method on Triangulate Domains), isometric signature, classification, geodesic distance.

1 Introduction

Automatic matching of nonrigid surfaces like articulated objects is a challenging problem in the field of shape and surface analysis. One of the milestones in this field was reported by Schwartz et al. in [32], where a multidimensional scaling (MDS) method was used to flatten a cortical triangulated surface onto a plane. This approach finds a uniform parameterization for convoluted surfaces, that can serve as a first step in a more general surface matching procedure. This early solution to the uniform surface parameterization problem led to recent efficient algorithms for the texture mapping problem in computer graphics [37] and to voxel-based flattening of the cortex in medical image analysis [20].

When matching curved surfaces, a better approach needs to be thought of. It has been recently shown that the high error introduced by restricting curved surfaces to a plane actually prevents the matching process of nontrivial objects [13]. We here propose to extend the "flat embedding" idea of Schwartz et al. and use higher dimensional Euclidean spaces to capture the intrinsic geometric structure of isometric surfaces. In our approach, the dimension of the embedding space is determined by the complexity of the surface in hand. By adding a few more dimensions and by embedding a given surface in a small dimensional space (rather than just a plane), we are able to go beyond the surface parameterization problem and design an accurate and robust matching procedure for convoluted surfaces. An important example is face recognition as reported in [13].

In computer vision, the problem of determining the similarity between two shapes has been well studied, see [2], [27], [30] for some shape matching solutions. Most existing techniques address rigid object matching by algorithms that search for the transformation that maximizes shape similarities while aligning the two objects, see e.g., [31]. Other examples are [7], [8], where Besl proposed metrics for measuring matches between curves and surfaces. These techniques were designed to handle rigid transformation and thus find it unnatural to deal with "nonrigid" or "articulated" objects.

Faugeras [18], and Faugeras and Hebert [19] 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 [26] solved 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 [6] associate a *footprint* for each surface point in order to separately extract the rotation and translation components of the desired rigid transformation. Again, these techniques focus on matching rigid objects. Surveys on these techniques can be found in [6], [10].

One less "rigid" technique to match shapes is by comparing their geometric-statistical properties. The idea is to compare discrete histograms of geometric measures. Thacker et al. [1], [5], [3], [4], and Horn [22] presented shapes in images by histograms of either differential measures, like normals or curvatures, or semilocal structures, like angles and distances between pairs of line segments. Besl [9] used the histogram of the crease angle for all edges in a 3D triangular mesh to assign a signature

- *A. Elad (Elbaz) is with ioIMAGE Ltd., 3 Maskit Street, Herzliya Pituach, PO Box 12414, Israel 46733. E-mail: asi@ioimage.com.*
- *R. Kimmel is with the Department of Computer Science, Technion–Israel Institute of Technology, Technion City, Haifa 32000, Israel. E-mail: ron@cs.technion.ac.il.*

Manuscript received 6 Feb. 2002; revised 19 Nov. 2002; accepted 29 Apr. 2003.
Recommended for acceptance by S. Dickinson.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number 115838.

0162-8828/03/$17.00 © 2003 IEEE Published by the IEEE Computer Society

for a given mesh. In a more recent work, Osada et al. [29] used a variation of this concept by evaluating the probability distribution function (PDF) of a given measure on a 3D surface such as the Euclidean distances between two random points on the surface. Then, they compared two surfaces by evaluating the distance between their statistical signatures. This technique is simple and somewhat robust to small perturbations. However, its classification rate is considered to be low.

Another set of shape matching techniques are based on comparing high level components of a given shape. Such a model-based approach first decomposes the given shape into primitive parts or set of features. Then, it measures similarity between different shapes based on their primitive parts (or sets of features). Bloomenthal and Lin [11] and Storti et al. [34] used such a method to construct a skeletal-based modeling for shape matching. These techniques work on rigid shapes that can be modeled by a skeletal representation.

In this paper, we extend ideas we put forward in [15] and introduce a different solution to a different surface matching problem. While most of the previous methods dealt mainly with rigid transformations or rely on key points and local or semidifferential invariant measures, here we address the bending invariant problem by a transformation that takes isometric surfaces to similar surfaces in a finite dimensional (flat) Euclidean space. In the Euclidean space, these surfaces can be compared using simple rigid object matching techniques. Our technique is designed to compare between full 2D surfaces as well as patches of surfaces.

Our method is based on two numerical procedures. The first is the *fast marching on triangulated domains* (FMTD) [23] that efficiently calculates geodesic distances on triangulated curved surfaces. The second is the Multi-Dimensional Scaling (MDS) [14], [12], [25] that reveals the geometric structure of a set of data items from the (dis)similarity information between them.

The outline of this paper is as follows: Section 2 gives an overview of the proposed approach and its various components. Section 3 presents the basic concepts of MDS and compares between three MDS methods: The *Classical Scaling*, the *Least-Squares*, and the *Fast MDS*. Clustering results for a few objects are presented in Section 4. Section 5 concludes the paper with comments on possible extensions like generating local signatures around key points for semilocal signatures.

2 Isometric Signatures

"An allowable mapping of the surface S onto the surface $\tilde{S}$ is said to be *isometric* or *length preserving* if the length of any arc on $\tilde{S}$ is the same as that of its inverse image on S" [24]. Our goal is to extract from this definition a numerical algorithm for matching two isometric surfaces or, in other words, surfaces for which the geodesic distances between corresponding surface points are the same. Therefore, our method should utilize intrinsic geodesic distances between surface points.

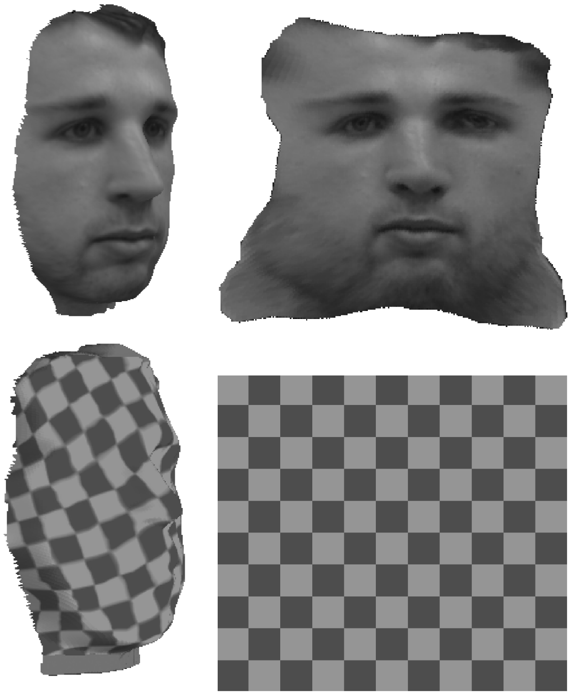

Fig. 1. Texture mapping—flattening via MDS. Top left: Original surface. Top right: Flattened surface. Bottom right: Texture on a plane. Bottom left: Mapping the texture on the original surface.

The proposed technique is a combination of two numerical schemes: the *fast marching on triangulated domains* (FMTD) algorithm and one of the *multidimensional scaling* (MDS) procedures. We use the geodesic distances between surface points that are invariant to surface bending. Then, by feeding a matrix of geodesic distances between surface points as a dissimilarity measure to an MDS process, we map isometric (bending-invariant) surfaces into similar signatures.

As mentioned in the introduction, similar steps were proposed by Schwartz et al. in [32] for surface flattening into a plane. Zigelman et al. [37] revisited the numerical ingredients of that framework and used an MDS flattening approach to efficiently solve the texture mapping problem. The MDS output space was set to $m = 2$, which means flattening the given surface onto a plane. Then, by using the correspondence between the points on the original surface and its flattened version, they mapped planar textures back to the surface with minimal local and global distortions, see Fig. 1. In a related paper [20], the FMTD procedure was replaced by an unbiased graph search-based technique and applied to voxel surfaces.

Here, by allowing more dimensions, we extend the above flat-embedding idea and map a given surface into a new one embedded in a small dimensional space (which is not restricted to be a plane). Let us start with a simple description of the main steps of the proposed method. Fig. 2 is a sketch of the process of computing the bending invariant signature surface for two different postures of a hand. The first step involves the computation of geodesic distances between points on the surface. On the second row of Fig. 2, one minimal geodesic path connecting two surface points is plotted as a thick white curve, while equal geodesic distance contours from one point are plotted as thinner white contours. The third row presents the bending invariant surfaces generated by transforming the geodesic distances between points into Euclidean ones (via an MDS procedure). This way, both hands are mapped into similar

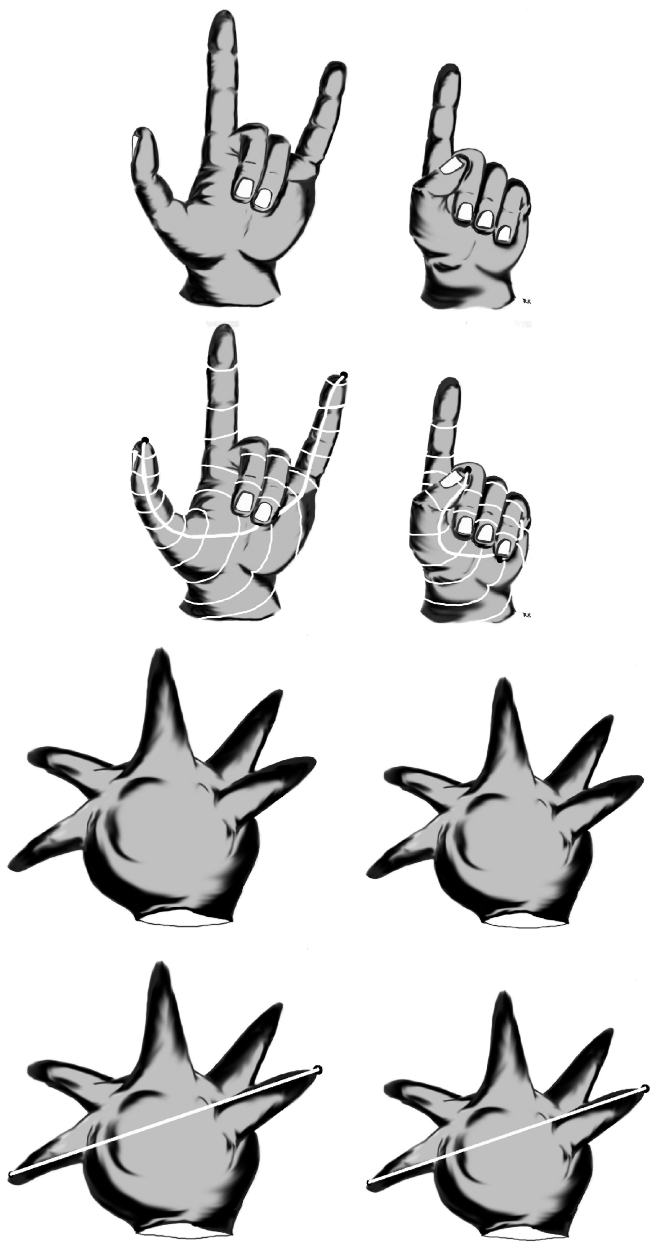

Fig. 2. Top row: Two different configurations of a hand. Second row: A minimal geodesic path connecting two surface points is plotted as a thick white curve, while equal geodesic distance contours are plotted as thinner white contours. Third row: Signature generated by transforming geodesic distances into Euclidean ones. Bottom row: The minimal geodesic curve on the hand becomes a straight line connecting the two corresponding points on the bending invariant surface.

looking surfaces. We use these bending invariant surfaces as signatures. At the bottom, the curve corresponding to the minimal geodesic connecting the two points on the original hands is transformed into a straight line connecting the two corresponding points on the signature surface.

In this paper, we deal with triangulated surfaces, though the framework can be applied to any kind of surface representation. We first sample the vertices of the triangles that describe the surface and select a subset of n points uniformly distributed on the surface. Next, we construct a matrix of geodesic distances between these n selected points on the surface. This way, we fill in the entries of an $n \times n$ symmetric dissimilarity matrix D with zeros along the diagonal. Based on an analysis of the surfaces we would like to match, we compute the MDS output dimension, m, and sampling rate, n. The output dimension m can be sensitive to the representation error, according to which m is increased as long as the representation error (that will be defined in Section 3) does not get below a given threshold. The MDS procedure produces a set of coordinates in a Euclidean space $\mathbb{R}^m$ that minimize a representation error. The Euclidean distances between the output points are as close as possible to the geodesic distances between the original coordinates. The MDS actually finds an embedding of the surface in the finite dimensional Euclidean space in which the geodesic distances are mapped to Euclidean ones. We denote this surface embedded in the Euclidean space as a bending invariant signature surface or as an isometric signature of the original surface.

2.1 Geodesic Distances

A recent paper by Hilaga et al. [21] also explores the intergeodesic distances between points on a surface in order to match nonrigid surfaces. The authors first define a scalar function on the surface they refer to it as the "distribution of geodesic distances." For each surface point, the "distribution function" is computed by integrating the geodesic distances from the given point to the rest of the points on the surface. Next, the skeletal structure of that scalar function is constructed and analyzed via multidimensional Reeb graphs (MRG). The similarity between surfaces is measured by matching their MRGs. This method is aimed at matching topologically similar surfaces rather than isometric surfaces. Actually, in their implementation, the geodesic distances on a triangulated surface were measured by the Dijkstra graph search algorithm, which is a numerically inconsistent approach that induces metrication errors if the surface is sampled in a regular way.

In order to compute the geodesic distances between pairs of points on the surface, we use another method, the *fast marching on triangulated domains* introduced by Kimmel and Sethian in [23]. It is an extension of Sethian's fast marching method [33], which is a numerically consistent distance computation algorithm that works on rectangular grids (see Tsitsiklis [36] Eikonal solver on rectangular grids for a related approach).

The basic idea is an efficient numerical approach that solves an Eikonal equation on the triangulated surface. The solution is a surface distance function that is proven to converge to the "viscosity" smooth solution as the numerical grid (triangulation) is refined. Like the Dijkstra graph search method, the distance function is constructed by starting from a sources point and propagating outwards.

The fast marching on triangulated domains method can compute the geodesic distance between one vertex and the rest of the $\tilde{n}$ surface vertices in $O(\tilde{n})$ operations. Repeating this computation for n $(n < \tilde{n})$ selected vertices, we can compute a geodesic distance matrix D in $O(n\tilde{n})$ operations. Each ij entry of D represents the square geodesic distance between vertex v_i and vertex v_j. That is,

$$\begin{aligned}\delta_{ij} &= d_S(v_i, v_j)\\ [D]_{ij} &= (\delta_{ij})^2,\end{aligned} \tag{1}$$

where $d_S(v_i, v_j)$ is the geodesic distance (measured on the surface S) between the surface point indicated by the vertex v_i, and the surface point indicated by the vertex v_j.

2.2 Triangulation, Complexity, and Accuracy

A triangulated surface is an approximate representation of a continuous one. The representation error introduced by triangulating a smooth surface is of an order of the length of the edges of the triangles. The smaller the triangles get, the more accurate the triangulation represents the surface. Obviously, the same argument holds when subsampling a triangulated surface. For computational efficiency of our numerical procedures, we would like to consider as small as possible a subset of the given initial vertices. In order to be able to distinguish between bending invariant signatures of nonisometric surfaces when sampling the surfaces, one should be aware of the relative size of the characteristic features differentiating the surfaces. The approximation step should be verified to keep objects that belong to different classes well separated.

Recall that, given surfaces in 3D, we would like to measure their isometric dissimilarity and thereby classify them. At the first step, we compute the geodesic distance matrix for each surface using the fast marching on triangulated domains. Given a triangulated surface with $\tilde{n}$ vertices, we select a subset of $n \leq \tilde{n}$ vertices and calculate the geodesic distances between each pair of vertices in this set using the original surface. The subsampling technique we used is an iterative process where, at each iteration, the farthest vertex (in geodesic sense) from the already selected ones is selected. The sampling process starts by randomly selecting the first vertex and terminates when the number of selected vertices reaches a predefined number. This subsampling procedure is also known as iterative Voronoi sampling, see, e.g., [16]. Vertices are chosen iteratively in a greedy manner: Each iteration the vertex with the largest geodesic distance to the rest of the chosen vertices is selected. If required, the geometric connectivity between the selected vertices is then computed by Melax triangulation decimation technique [28].

2.3 MDS-Mapping and Matching

The MDS-mapping step involves setting the required dimension, m, and an application of one of the MDS techniques described in Section 3 on the geodesic distances matrices. The MDS technique produces new coordinates of the surface vertices that we named isometric signature. We can also handle similarity (uniform scaling) transformations by normalizing the signature surface. After the MDS-mapping, the signature surfaces are uniformly scaled into a unit bounding box, centered (automatically in classical MDS), and oriented using the second order moments (the eigenvalues in the classical MDS).

Finally, we have to compare between the signature surfaces that are given by their coordinates in $\mathbf{R}^m$. We can construct a distance matrix between these given surfaces based on some unique measure, like the Hausdorff distance. Actually, any algorithm that finds a similarity between rigid objects can be used at this stage of our framework. We have practically transformed the problem of matching isometric/nonrigid/articulated surfaces into the problem of matching "rigid objects" that we refer to as isometric signatures, with the following properties:

- The dimension of the embedding space in which these invariant surfaces or rigid objects "live" depends on the complexity of the original surface.
- When using classical scaling, the second order cross moments vanish.
- The orientation alignment about the center of mass is also solved as a by-product of classical scaling.

Let us list some candidate measures for the rigid object matching step of of our framework.

- Apply heuristic algorithms that search for the transformation that maximizes shape similarities while aligning the two objects [31], [7], [8].
- Evaluate the probability distribution function (PDF) of a given geometric measure on a 3D surface such as the Euclidean distance between two random points on the surface as proposed in [29].
- Use a simple model-based approach to decompose the given shapes into parts or set of features and then compute (heuristically) a dissimilarity measure between different shapes based on those parts (or sets of features) [11], [34].

A relatively simple matching measure for rigid objects was proposed by Tal and Elad in [35]. They consider only the first few moments (less than 15) of each surface and calculate the Euclidean distances between each pair of such moments vectors. We use this simple concept to demonstrate that the main clustering work was done by generating the bending invariant signatures and, from there on, any trivial classifier will do.

Let M_i be a vector of the first few moments of the surface S_i, then the moments-distance matrix, D_M, is defined as

$$[D_M]_{ij} = \|M_i - M_j\|_2^2.$$

Applying again an MDS procedure on this matrix yields points in a Euclidean space, where each point represents one surface. We expect signatures of isometric surfaces to be clustered together while signatures of nonisometric surfaces will be well separated in this new Euclidean space, as will be illustrated in Section 4.

3 Multidimensional Scaling Methods

Multi-Dimensional Scaling (MDS) is a family of methods that map 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 an MDS procedure enables us to view the data and explore its geometric structure. Most metrical MDS methods expect a set of n items and their pairwise (dis)similarities and the desired dimensionality, m, of the Euclidean embedding space. As MDS is an important step in our framework, in this section, we briefly review three

MDS methods and evaluate their properties which are relevant to our framework.

MDS algorithms map each item to a point $x_i = X_i$ in an m-dimensional Euclidean space $\mathbf{R}^m$ by minimization of, for example, the stress function,

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

where δ_{ij} is the input dissimilarity measure between item i and j, $d_{ij}(X)$ is the Euclidean distance between these items in the m-dimensional Euclidean space, and w_{ij} are some weighting coefficients. In our problem, as proximity values we use the geodesic distances computed by the *fast marching on triangulated domains* procedure.

Let us start with a simple example in which we fix $m = 3$. Given the dissimilarities matrix, an MDS procedure produces coordinates, X, in $\mathbf{R}^3$, for which the Euclidean distances between the points in $\mathbf{R}^3$ would be as close as possible under some norm 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 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-mapping and obtain a surface that we refer to as a *bending invariant signature* of the original surface.

The selection of $\mathbf{R}^3$ in this example was arbitrary. In order to select a "proper" dimension, m, we define the effective dimensionality of the problem to be m, the smallest number of dimensions that allow an invariant signature to capture a given percentage of the representation energy. In other words, the representation error captured by the Euclidean embedding in $\mathbf{R}^m$ is smaller than a predefined threshold.

In our experiments, we allowed a representation error of up to 5 percent. For $m = 3$, we can graphically view the invariant signature as a surface in $\mathbf{R}^3$. For $m \leq 6$, the invariant signature can be viewed as a colored surface, where the first three coordinates are used as physical coordinates in $\mathbf{R}^3$ and the next coordinates are used as color indices. Actually, as defined above, m itself is an invariant measure that captures both the topological and topographic complexity of the surface.

3.1 Classical MDS

Classical scaling was originated in the 1930s when Young et al. showed that, given a matrix of distances between points in a Euclidean space, one can extract coordinates such that distances are preserved, see, e.g., [12]. This method allows us to evaluate, in a simple way, the representation error for a given surface in a given number of dimensions.

Let the coordinates of n points in a k dimensional Euclidean space $\mathbf{R}^k$ be given by x_r, $(r = 1, \ldots, n)$, where $x_r = [x_{r1}, x_{r2}, \ldots, x_{rk}]^T$. The Euclidean distance between the rth and the sth points is given by

$$d_{rs}^2 = [x_r - x_s]^T [x_r - x_s]. \tag{3}$$

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 [14])

$$B = -\frac{1}{2} JDJ, \tag{4}$$

where

$$J = I - \frac{1}{n}\mathbf{1}\mathbf{1}^T$$
$$\mathbf{1}_{1\times n} = [1, 1, \ldots, 1]^T.$$

We also have that $B = XX^T$, where $X = [x_1, \ldots x_n]^T$ is the $n \times k$ matrix of the coordinates. The inner product matrix B is symmetric, positive semidefinite, and of rank k. Therefore, B has k nonnegative eigenvalues and $n - k$ zero eigenvalues. The matrix B can be expressed in terms of its spectral decomposition,

$$B = V \Lambda V^T, \tag{5}$$

where

$$\Lambda_{n\times n} = \mathrm{diag}(\lambda_1, \lambda_2, \ldots, \lambda_k, 0 \ldots, 0).$$

For convenience, the eigenvalues of B are ordered such that $\lambda_1 \geq \lambda_2 \geq \ldots \lambda_k \geq 0$. Hence, the required coordinates are given by using the nonzeros submatrix $\Lambda_{k\times k}$ and the corresponding eigenvectors submatrix $V_{n\times k}$,

$$X_{n\times k} = V_{n\times k} \Lambda_{k\times k}^{\frac{1}{2}}.$$

The classical scaling is considered to be an efficient algebraic approach to solving 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 their corresponding eigenvectors, which can be computed by variations of the "power method," see, for example, [25]. Instead of the stress function (2), the classical MDS approach minimizes a version of the Frobenius norm given by

$$E = \| V \cdot (\Lambda - \widetilde{\Lambda}) \cdot V^T \|, \tag{6}$$

where

$$\widetilde{\Lambda} = \mathrm{diag}(\lambda_1, \lambda_2, \ldots, \lambda_m, 0 \ldots 0),$$
$$\Lambda = \mathrm{diag}(\lambda_1, \lambda_2, \ldots, \lambda_m, .., \lambda_k, 0 \ldots 0),$$

and $m \leq k$.

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 (2). The problem is that there is no simple way to form a closed expression for the first derivative of this nonlinear functional. A 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 the stress [12]. 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, $\hat{S}(X, Z) \geq S(X)$ for

TABLE 1
LS MDS: The Stress (2) as a Function of the Number of Dimensions m

Obj/Dim	$m = 1$	$m = 2$	$m = 3$	$m = 4$	$m = 5$
Torus	0.325	0.0714	0.0098	0.0098	0.0098
Human	0.323	0.0341	0.0051	0.0051	0.0051
Rabbit	0.345	0.044	0.0029	0.0021	0.0021
Elephant	0.316	0.113	0.0043	0.0036	0.0036

$X \neq Z$, and $\hat{S}(Z, Z) = S(Z)$. Let us briefly review this procedure. For further details, see [12].

Minimizing the stress (2) is equivalent to minimizing the following functional.

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

or

$$S(X) = \eta_\delta^2 + \eta(X)^2 - 2\rho(X),$$

where

$$\eta_\delta^2 = \sum_{i<j} w_{ij} \cdot \delta_{ij}^2,$$
$$\eta(X)^2 = \sum_{i<j} w_{ij} d_{ij}(X)^2,$$
$$\rho(X) = \sum_{i<j} w_{ij}\delta_{ij} \cdot d_{ij}(X),$$

and

$$d_{ij}(X) = \left(\sum_{a=1}^{n} (x_{ia} - x_{ja})^2\right)^{1/2}.$$

Applying *Cauchy-Schwartz* inequality and basic algebraic operations, it can be shown that the stress function can be bounded by the following quadratic function in X:

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

where the matrix $B(Z)$ elements are

$$b_{ij} = \begin{cases} -\frac{w_{ij}\delta_{ij}}{d_{ij}(Z)} & i \neq j \text{ and } d_{ij}(Z) \neq 0 \\ 0 & i \neq j \text{ and } d_{ij}(Z) = 0 \end{cases}$$
$$b_{ii} = \sum_{j=1, j\neq i}^{n} b_{ij},$$

and the matrix V elements are given by

$$V_{ij} = \sum_{i<j} w_{ij} A_{ij}$$
$$A_{ij} = (e_i - e_j)(e_i - e_j)^T,$$

where e_i is the ith column of the identity matrix and thereby A_{ij} is an $n \times n$ matrix.

The minimum of $\tau(X, Z)$ can be extracted by setting the derivative of $\tau(X, Z)$ to zero and the required solution is given by

$$X_i = V^+ B(Z) Z. \tag{8}$$

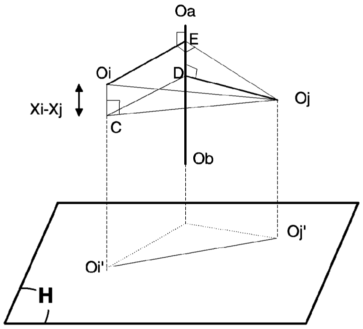

Fig. 3. Projection on the hyperplane H.

Here, 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

$$X_i = n^{-1} B(Z) Z. \tag{9}$$

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 (8) or (9).
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 1, decreases the stress (2) by less than 1 percent in our examples. In all of our test cases, the LS MDS required less than a hundred iterations to converge. Hence, the complexity is of $O(n^2 \cdot \mathrm{NumOfIterations})$.

3.3 Fast MDS

The Fast MDS is a recent heuristic technique proposed by Faloutsos and Lin [17]. This method is computationally efficient, $O(nm)$ where m is the target dimension, that can be considered to be $O(n)$ in our case. Yet, unlike the two previous procedures, it does not minimize any global measure, but merely 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 axis. First, the algorithm selects two "pivot" vertices, O_a and O_b, that are as far as possible from one other. Next, all other vertices are projected on the line defined by (O_a, O_b) using the *cosine law* $x_i = \frac{d_{ai}^2 + d_{ab}^2 - d_{bi}^2}{2d_{ab}}$, see Fig. 3.

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 generate a new distance matrix according to,

$$d_{i'j'}^2 = d_{ij}^2 - (x_i - x_j)^2. \tag{10}$$

This step should be repeated m times. At each step, the calculated x_i, $i = (1, 2 \ldots, n)$, are the new added dimension coordinates. The m-dimensional coordinates can be

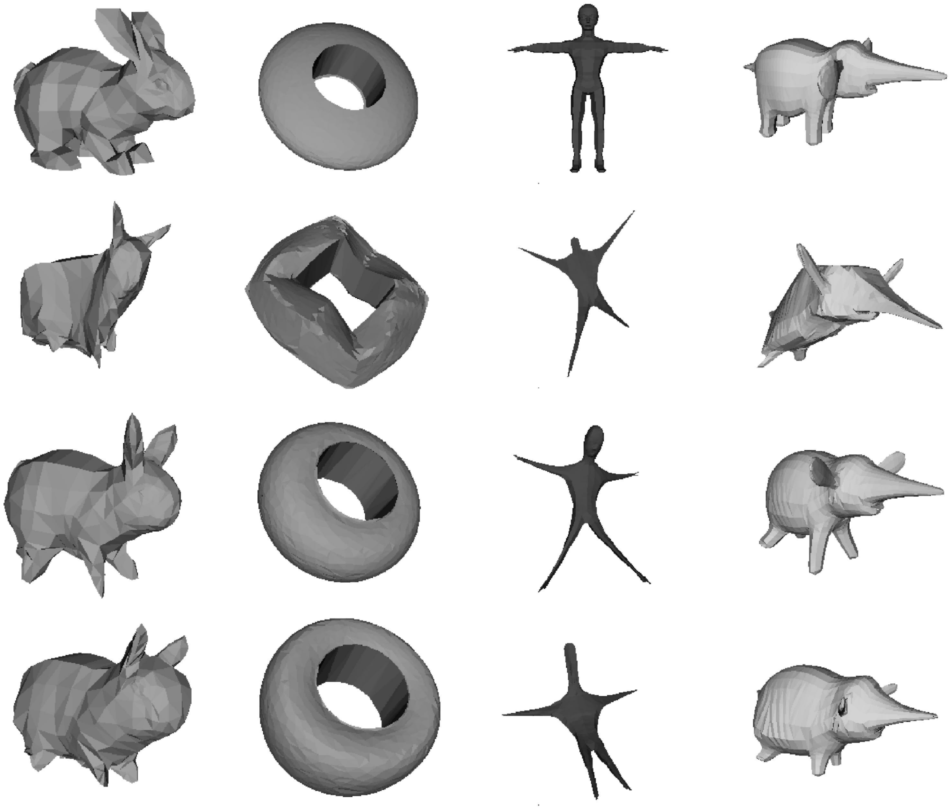

Fig. 4. Top row: Original surface. Second row: Fast MDS result. Third row: LS MDS result. Bottom row: Classical MDS result.

calculated in $O(mn)$ provided that, at each step, we use a linear heuristic algorithm to choose the two pivot vertices.

3.4 Evaluating the MDS-Mappings

We computed invariant signatures of various surfaces by the above three MDS techniques and obtained the results in Fig. 4 for $m = 3$. Table 2 summarizes the stress (2) for increasing dimensions of the three MDS techniques. As expected, the heuristic fast method produced the least accurate results (in terms of stress) yet was the fastest, while the accurate (in terms of stress) LS technique was the slowest. The LS method reaches the minimal stress, which is not surprising since the classical MDS minimizes another measure (the Frobenius norm) as described above. Nevertheless, we can see that the stress decreases for all three methods as the number of dimensions, m, gets larger.

TABLE 2
The Stress (2) for LS, Classical, and Fast MDS as a Function of m, the Number of Dimensions of the Embedding Euclidean Space

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	0.325	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	0.323	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	0.345	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	0.316	0.113	0.0043	0.0036	0.0036
Fast	13.28	0.3051	0.0381	0.0122	0.0077

Our next goal is to evaluate the various MDS procedures as part of the whole classification framework. In our examples the LS method that minimizes the stress function better captures the geometric structure of the data compared to the

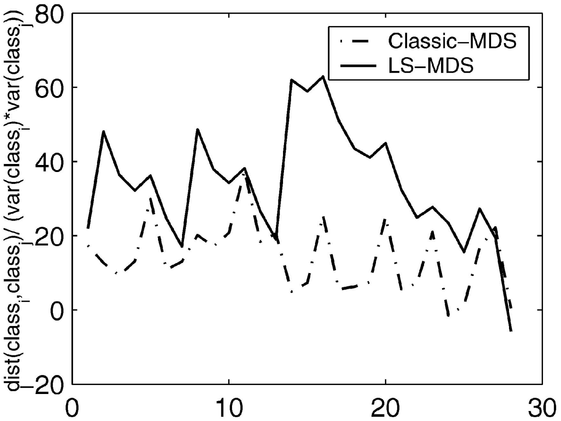

Fig. 5. The Log-ratio of the distances between two classes of two different objects and the variances of the classes. The solid line represents the LS and the dashed line the classic MDS.

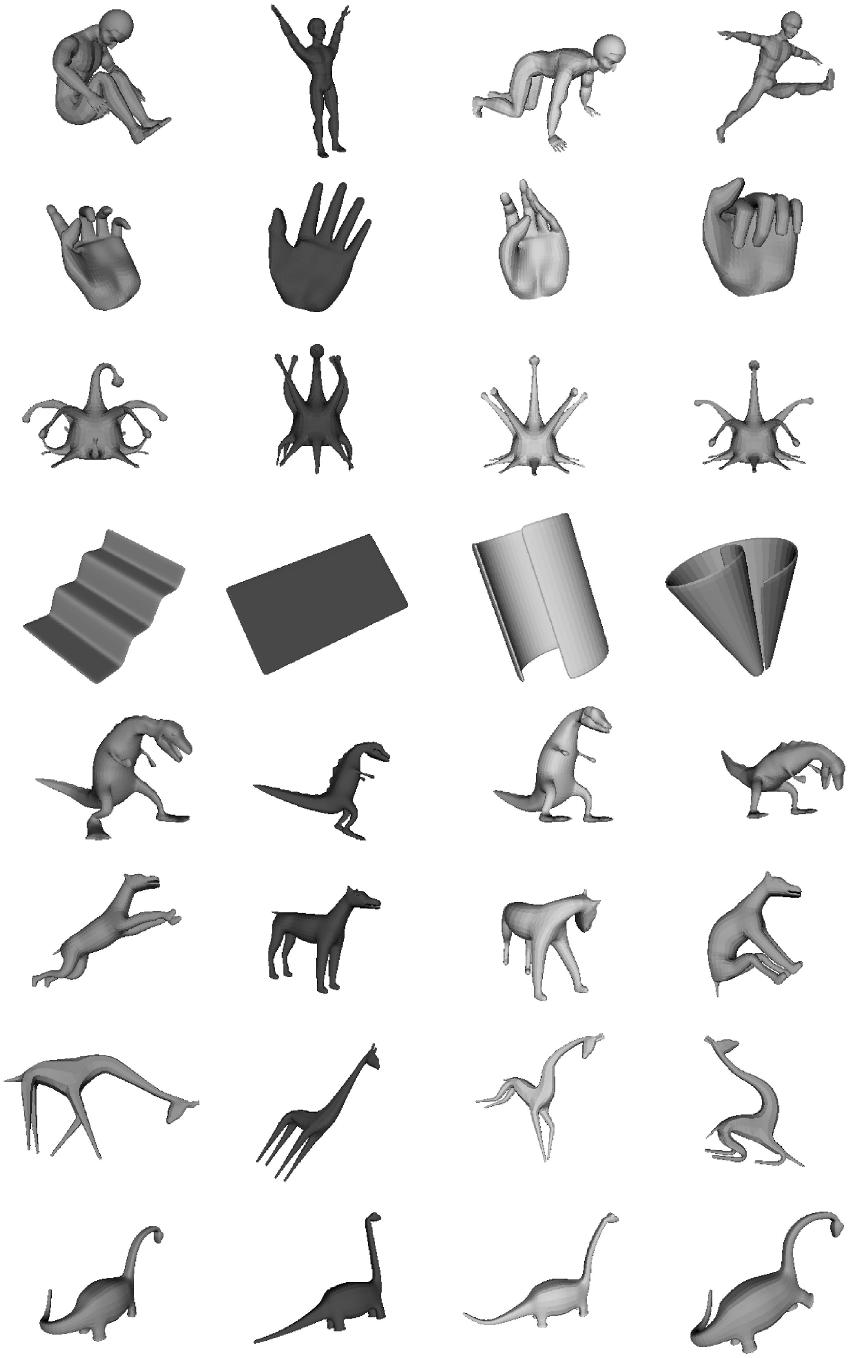

Fig. 6. Eight input surfaces of two kinds of dinosaurs (Barosaurus and Tyrannosaurus Rex), a human body, a hand, a hat, a paper, a dog, a giraffe, and a few bending versions of these surfaces.

results obtained by classical scaling and the fast MDS. That is, isometric surfaces were better clustered and better separation between the different classes was obtained by minimizing the stress. Fig. 5 shows two graphs of the ratio between the distance between the centers of mass of two classes, and the variances of these classes. The better the separation, the higher these ratios should be. In most of our examples, the LS gave better separation results, as shown by the solid black curve.

The above analysis leads to the almost obvious conclusion, that the faster the MDS method, the worse its performances in terms of stress and separation between classes.

4 Classifying Isometric Surfaces

We applied the proposed algorithm to the surfaces shown in Fig. 6. The input includes bending versions of six different surfaces, A human body, a hand, a hat, a paper, a dog, two dinosaurs, and a giraffe. For the MDS-mapping results shown in Fig. 7 we used the Least Squares MDS. For presentation purposes, justified by an error analysis of our 3D objects, we selected the Euclidean embedding space to be of three dimensions. One can see that isometric surfaces are mapped to similar geometric structures.

The results of applying the moments-based clustering step is shown in the right frame of Fig. 8. As a reference, the

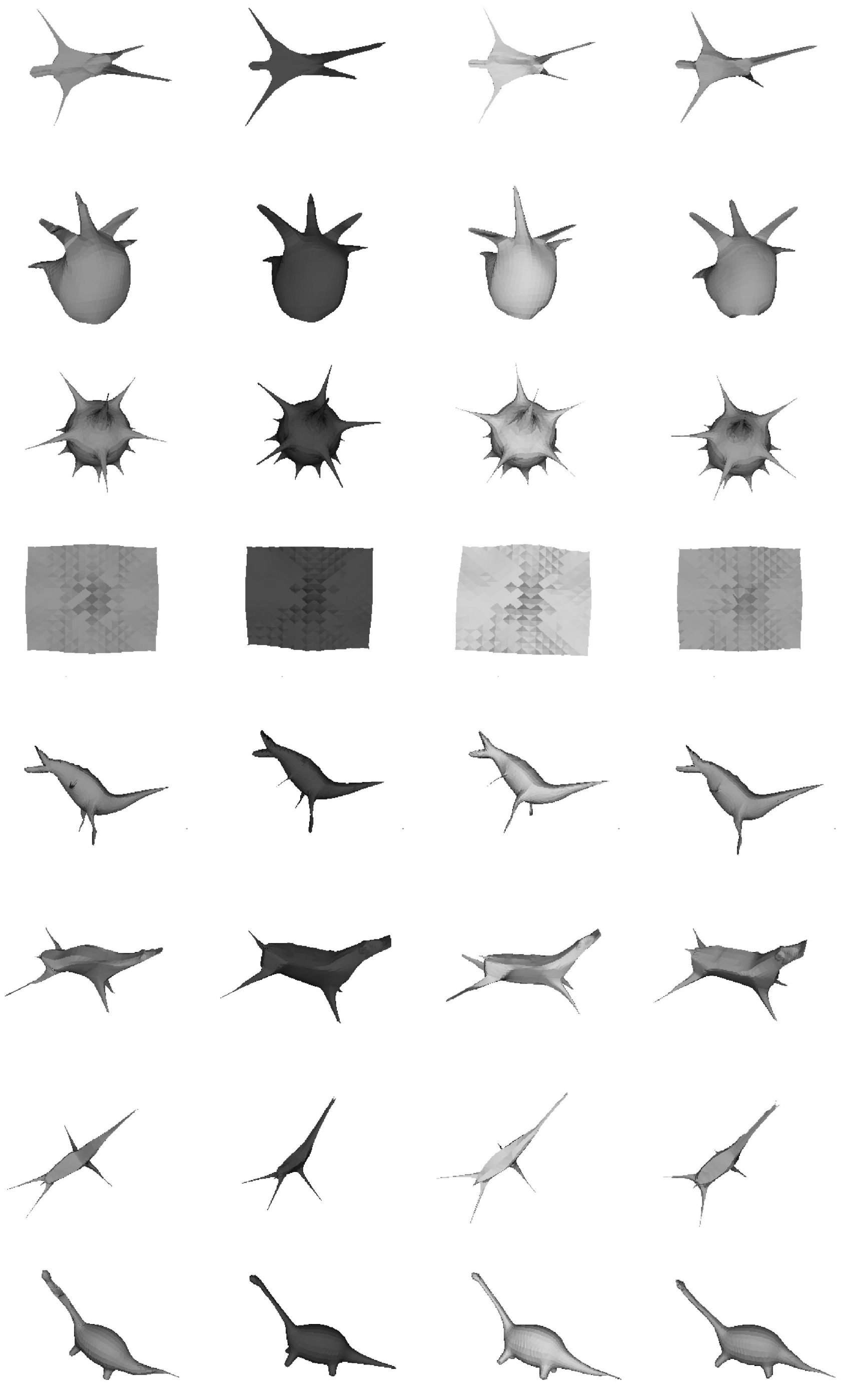

Fig. 7. Output—the invariant signature surfaces using LS MDS.

left frame displays the results of applying the moment-based MDS clustering procedure to the original surfaces. The isometric signature surfaces form small and compact well-separated groups compared to the moments-based clustering of the original surfaces. Again, the moments-based distances between the invariant signatures and final MDS-based clustering and classification steps are just one example for comparing the bending invariant signatures of the surfaces. It has been chosen mainly because it is similar in nature to the first steps of constructing the invariant signatures, which is the main focus of this paper. Other matching techniques between rigid objects that operate on the invariant signature surfaces are possible and would most probably yield better clustering results.

5 Conclusions

An efficient method for computing bending invariant signatures of isometric surfaces was presented. The method is based on the fast marching on triangulated domains algorithm followed by a multidimensional scaling (MDS) technique. An invariant signature of a surface is computed by applying an MDS procedure on the geodesic distances

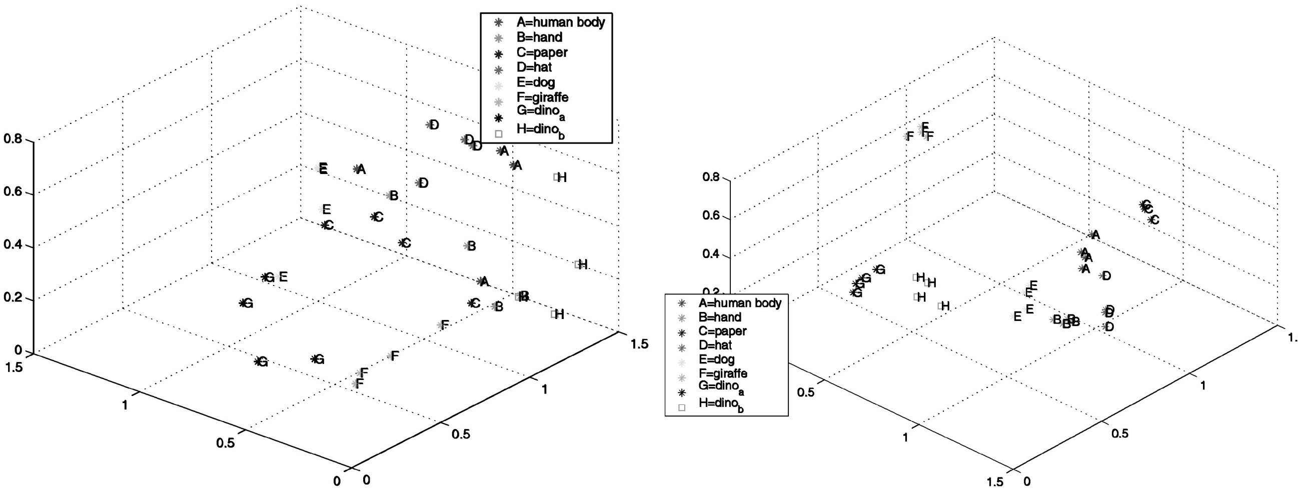

Fig. 8. Applying the moments-based clustering to the original surfaces (left) and to their bending invariant representations (right).

matrix computed by the fast marching on triangulated domains algorithm. Three different MDS techniques were tested and evaluated for this task: the Classical, the Least Squares, and the Fast MDS. Our approach, followed by a simple clustering algorithm, was shown to be useful for nonrigid isometric surface classification.

Semilocal invariant signatures can help in identifying nonrigid objects that are partially occluded. A possible extension of the method to handle partially occluded surfaces is a local application of the signature around key surface points. The neighborhood of the key points is defined by geodesic circles around the key points on the surface, while the key points themselves should be selected by analyzing isometric invariant measures, like extremum points of the Gaussian curvature. One such application for face recognition is currently under development with promising preliminary results [13].

Acknowledgments

The authors would likt to thank Gil Zigelman from 3DV-systems and Nahum Kiryati from Tel-Aviv University for interesting discussions and for allowing us to use Fig. 1. This research was partly supported by Technion V.P.R. fund—Venezuela Technion Society Research fund and by the fund for the Promotion of Research at the Technion.

References

[1] F.J. Aherne, N.A. Thacker, and P.I. Rockett, "Optimal Pairwise Geometric Histograms," *Proc. British Machine Vision Conf. (BMVC),* pp. 480-490, 1997.

[2] F. Arman and J.K. Aggarwal, "Model-Based Object Recognition in Dense-Range Images—A Review," *ACM Computing Surveys,* vol. 25, no. 1, pp. 5-43, 1993.

[3] A.P. Ashbrook, N.A. Thacker, and P.I. Rockett, "Multiple Shape Recognition using Pairwise Geometric Histograms Based Algorithms," *Proc. IEEE Image Processing,* 1995.

[4] A.P. Ashbrook, N.A. Thacker, and P.I. Rockett, "Pairwise Geometric Histograms: A Scalable Solution for Recognition 2D Rigid Shapes," *Proc. Ninth Scandinavian Conf. Image Analysis,* vol. 1, pp. 271-278, 1995.

[5] A.P. Ashbrook, N.A. Thacker, P.I. Rockett, and C.I. Brown, "Robust Recognition of Scaled Shapes Using Pairwise Geometric Histograms," *Proc. British Machine Vision Conf. (BMVC),* pp. 503-512, 1995.

[6] G. Barequet and M. Sharir, "Partial Surface and Volume Matching in Three Dimensions," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 19, no. 9, pp. 929-948, Sept. 1997.

[7] P.J. Besl, "Geometric Modeling and Computer Vision," *Proc. IEEE,* vol. 76, pp. 936-958, 1988.

[8] P.J. Besl, "The Free Form Matching Problem," *Machine Vision for Three-Dimensional Scene,* H. Freeman, ed., New York: Academic, 1990.

[9] P.J. Besl, "Triangles as a Primary Representation," *Object Recognition in Computer Vision,* pp. 191-206, 1994.

[10] P.J. Besl and R.C. Jain, "Three-Dimensional Object Recognition," *ACM Computing Surveys,* vol. 17, no. 1, pp. 75-154, 1985.

[11] J. Bloomenthal and C. Lin, "Skeletal Methods of Shape Manipulation," *Proc. Shape Modeling and Applications,* pp. 44-47, 1999.

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

[13] A. Bronstein, M. Bronstein, and R. Kimmel, "Expression-Invariant 3D Face Recognition," Proc. Audio and Video-Based Biometric Person Authentication (AVBPA), pp. 62-69, 2003.

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

[15] A. Elad and R. Kimmel, "Bending Invariant Representations for Surfaces," *Proc. Computer Vision and Pattern Recognition (CVPR),* Dec. 2001.

[16] Y. Eldar, M. Lindenbaum, M. Porat, and Y.Y. Zeevi, "The Farthest Point Strategy for Progressive Image Sampling," *IEEE Trans. Image Processing,* vol. 6, no. 9, pp. 1305-1315, 1997.

[17] C. Faloutsos and K.D. Lin, "A Fast Algorithm for Indexing, Data-Mining and Visualisation of Traditional and Multimedia Datasets," *ACM SIGMOD,* pp. 163-174, May 1995.

[18] O.D. Faugeras, "New Steps towards a Flexible 3D Vision System for Robotics," *Proc. IEEE Seventh Int'l Conf. Pattern Recognition,* vol. 2, pp. 796-805, 1984.

[19] 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 Intelligence,* pp. 996-1002, Aug. 1983.

[20] R. Grossman, N. Kiryati, and R. Kimmel, "Computational Surface Flattening: A Voxel-Based Approach," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 24, no. 4, pp. 433-441, Apr. 2002.

[21] M. Hilaga, Y. Shinagawa, T. Kohmura, and T.L. Kunii, "Topology Matching for Fully Automatic Similarity Estimation of 3D Shapes," *Proc. SIGGRAPH '01,* pp. 203-212, 2001.

[22] B.K.P Horn, "Extended Gaussian Images," *Proc. IEEE,* vol. 72, no. 12, pp. 1671-1686, 1984.

[23] R. Kimmel and J.A. Sethian, "Computing Geodesic Paths on Manifolds," *Proc. Nat'l Academy of Science,* vol. 95, pp. 8431-8435, 1998.

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

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

[26] 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 and Machine Intelligence,* vol. 17, no. 4, pp. 378-390, Apr. 1995.

[27] S. Loncaric, "A Survey of Shape Analysis Techniques," *Pattern Recognition,* vol. 31, no. 8, pp. 983-1001, 1998.

[28] S. Melax, "A Simple, Fast and Effective Polygon Reduction Algorithm," *Game Developer J.,* Nov. 1998.

[29] R. Osada, T. Funkhouser, B. Chazelle, and D. Dobkin, "Matching 3D Models with Shape Distributions," *Int. Conf. on Shape Modeling and Applications,* pp. 154-166, 2001.

[30] A.R. Pope, "Model-Based Object Recognition: A Survey of Recent Research," Technical Report TR-94-04, Univ. of British Colombia, 1994.

[31] M. Potmesil, "Generation of 3D Surface Descriptions from Images of Pattern-Illuminated Objects," *Proc. IEEE Conf. Pattern Recognition and Image Processing,* pp. 553-559, 1979.

[32] E.L. Schwartz, A. Shaw, and E. Wolfson, "A Numerical Solution to the Generalized Mapmaker's Problem: Flattening Nonconvex Polyhedral Surfaces," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 11, no. 9, pp. 1005-1008, Sept. 1989.

[33] J.A. Sethian, "A Review of the Theory, Algorithms, and Applications of Level Set Method for Propagating Surfaces," *Acta Numerica,* 1996.

[34] D.W. Storti, G. Turkiyyah, M. Ganter, C. Lim, and D. Stal, "Skeleton Based Modeling Operations on Solids," *Proc. Third ACM Solid Modeling,* pp. 141-154, 1997.

[35] A. Tal and M. Elad, "Similarity between 3D Objects—An Iterative and Interactive Approach," *IEEE Trans. Neural Networks,* submitted

[36] J.N. Tsitsiklis, "Efficient Algorithms for Globally Optimal Trajectories," *IEEE Trans. Automatic Control,* vol. 40, pp. 1528-1538, 1995.

[37] G. Zigelman, R. Kimmel, and N. Kiryati, "Texture Mapping Using Surface Flattening via MDS," *IEEE Trans. Visualization and Computer Graphics,* vol. 8, no. 2, pp. 198-207, Apr.-June 2002.

Asi Elad (Elbaz) received the BSc degree (with honors) in electrical engineering in 1998 and the MSc degree in computer science in 2002 from the Technion-Israel Institute of Technology. His research interests are in image processing and analysis, real-time systems, computer vision, and computer graphics. He presented his research results at the CVPR 2001 conference in Hawaii. He worked at Intel, Israel, during the years 1996-2000 and has been working at ioIMAGE since 2002.

Ron Kimmel received the BSc degree (with honors) in computer engineering in 1986, the MS degree in 1993 in electrical engineering, and the DSc degree in 1995 from the Technion-Israel Institute of Technology. During the years 1986-1991, he served as an R&D officer in the Israeli Air Force. During the years 1995-1998, he was a postdoctoral fellow at the Computer Science Division of Berkeley Labs and the Mathematics Department, University of California, Berkeley. Since 1998, he has been a faculty member of the Computer Science Department at the Technion, Israel, where he is currently an associate professor. His research interests are in computational methods and their applications in differential geometry, numerical analysis, image processing and analysis, computer aided design, robotic navigation, and computer graphics. Dr. Kimmel was awarded the Hershel Rich Technion Innovation Award, the Henry Taub Prize for excellence in research, Alon Fellowship, the HTI Postdoctoral Fellowship, and the Wolf, Gutwirth, Ollendorff, and Jury fellowships. He was a consultant to HP Research Lab in image processing and analysis during the years 1998-2000 and to the Net2Wireless/Jigami research group during 2000-2001. He is a senior member of the IEEE.