Bending Invariant Meshes and Application to Groupwise Correspondences

Stefanie Wuhrer Chang Shu Jonathan Boisvert Guy Godin
Pengcheng Xi
National Research Council of Canada
Ottawa, Canada

Abstract

We introduce a new bending invariant representation of a triangular mesh S. The bending invariant mesh X of S is a deformation of S that has the property that the geodesic distance between each pair of vertices on S is approximated well by the Euclidean distance between the corresponding vertices on X. Furthermore, X is intersection-free. The main advantage of the bending invariant mesh compared to previous approaches is that mesh-based features on X can be used to facilitate applications such as shape recognition or shape registration. We apply bending invariant meshes to find dense point-to-point correspondences between a number of deformed surfaces corresponding to different postures of the same non-rigid object in a fully automatic way.

1. Introduction

Bending invariant representations of a shape S aim to represent the intrinsic geometry of S in an embedding space $\mathcal{S}$ with simple extrinsic geometry. This representation simplifies the task of comparing, matching [14], [7], or corresponding [19], [34], [10] two articulated shapes $S^{(1)}$ and $S^{(2)}$ of the same underlying topology, because the bending invariant representations of $S^{(1)}$ and $S^{(2)}$ are near-rigid transformations of each other if and only if $S^{(1)}$ and $S^{(2)}$ are similar. We consider the case where S is represented by a possibly incomplete triangular mesh.

Our goal is to find dense point-to-point correspondences between a population of s deformed shapes $S^{(1)}, \ldots, S^{(s)}$. That is, for each vertex $p^{(1)}$ of $S^{(1)}$, we aim to find the vertex $p^{(r)}$ of $S^{(r)}$ corresponding to the same intrinsic location on $S^{(r)}$ as $p^{(1)}$ on $S^{(1)}$ for $r = 2, \ldots, s$. We need to find dense point-to-point correspondences between a population of deformed shapes in order to perform shape comparison for this population. For this application, it is important that the correspondence is unbiased.

We assume that the deformation preserves the intrinsic geometry of the shape. That is, corresponding lengths measured on the surfaces $S^{(r)}, r = 1, \ldots, s$ are identical. Deformations with this property are called *isometric*. We relax this condition to allow for small changes in intrinsic geometry. This relaxed condition is a good approximation of the locomotion of human being and many other animals. When near-isometric deformations are considered, bending invariant representations of the s shapes can be used to facilitate computing the correspondences.

With this model, the problem of computing a bending invariant representation of S becomes an embedding problem: given a symmetric dissimilarity matrix Δ and a symmetric weight matrix W, find points X in $\mathcal{S}$, such that an embedding energy $E_{EMB}(\Delta, W, X)$ is minimized. The embedding energy $E_{EMB}(\Delta, W, X)$ aims to find an embedding, such that for each pair of vertices, the dissimilarity $\delta_{i,j}$ between two vertices on S and the distance $d_{\mathcal{S}}(\vec{x}_i, \vec{x}_j)$ in $\mathcal{S}$ between the corresponding embedding points are similar. The influence of a vertex pair can be weighted by the entry $\omega_{i,j}$ of W.

Bending invariant representations of this type are invariant with respect to rotation, translation, and reflections. Hence, to find point-to-point correspondences between two embeddings $X^{(1)}$ and $X^{(2)}$ in $\mathbb{R}^d$, we need to consider the 2^d possible alignments obtained by flipping the shapes with respect to the coordinate axes [19]. When we aim to find dense point-to-point correspondences between a population of s articulated shapes, the number of possible alignments becomes exponential in s. Even when all of these alignments are considered, symmetric misalignments may occur because the surface orientation is not maintained [19], [34]. This renders these bending invariant representations impractical for registering s articulated shapes.

Generalized multi-dimensional scaling [8] uses a triangular mesh as embedding space $\mathcal{S}$. This method is shown to perform well when registering two articulated shapes. However, since the orientation of the shape is not maintained, local surface flippings may occur. When the goal is to compute correspondences between a population of s articulated shapes, one of the shapes, say $S^{(1)}$, can be used as the embedding space. Embedding $S^{(r)}, r = 2, \ldots, s$ into $S^{(1)}$

gives the dense point-to-point correspondences between the given population. However, this introduces a bias towards $S^{(1)}$ in the result because the extrinsic geometry of embeddings depends on the extrinsic geometry of $S^{(1)}$. This is undesirable when the aim is to use the correspondences for shape analysis.

We introduce a new bending invariant representation of a shape S. The *bending invariant mesh X of S* is a deformation of S that has the following two properties. First, the geodesic distance between each pair of vertices on S is approximated well by the Euclidean distance between the corresponding vertices on X. Second, X has an intersection-free underlying mesh. These two properties imply that (a) the surface of X has the same orientation as the surface of S and that (b) the extrinsic geometry of X depends only on the intrinsic and extrinsic geometries of S.

To compute the bending invariant mesh X of a triangular mesh S with n vertices, we deform S using multi-dimensional scaling with an added repelling energy term.

When computing dense point-to-point correspondences between s deformed shapes, we use feature points on the bending invariant meshes to find a rigid registration. Starting from this initial alignment, we iteratively improve the correspondences using generalized Procrustes analysis. This allows an unbiased registration in polynomial time.

2. Related Work

To compute a bending invariant representation of a shape S, geodesic distances are commonly used to measure the dissimilarity between pairs of vertices on S. If holes are present in S, the geodesic distances are often weighted depending on whether or not they pass by a hole of S [28], [34]. In order to reduce the sensitivity of the dissimilarities with respect to topological noise, Bronstein et al. [11] use as dissimilarity a distance that combines the intrinsic and extrinsic distances between the shapes, and Bronstein et al. [5, Chapter 12] use as dissimilarity a diffusion distance approximating the average path length of all paths connecting pairs of vertices on S.

Bending invariant representations were first introduced by Elad and Kimmel [14]. The representation is called *canonical form* and it aims to find an embedding in $\mathcal{S} = \mathbb{R}^k$ for a positive constant k. In their work, three embedding algorithms based on multi-dimensional scaling (MDS) are tested. The first algorithm computes a canonical form via classical MDS [17]. Computing an embedding via classical MDS takes $O(kn^2)$ time [22]. The second algorithm computes a canonical form via least-squares MDS [4, p.146-155]. The algorithm takes $O(n^2t)$ time, where t is the number of iterations needed until convergence of X. The third algorithm computes a canonical form via Fast MDS [15]. The algorithm takes $O(kn)$ time.

Canonical forms were used for a variety of applications. Bronstein et al. [7] use canonical forms for expression invariant face recognition. Bronstein et al. [12] introduce multigrid MDS to achieve more efficient algorithms. Jain and Zhang [18] apply the surface recognition algorithm to general surfaces. They improve the efficiency of the algorithm by computing a form via the Nyström approximation. Jain et al. [19] and Wuhrer et al. [34] use canonical forms to find one-to-one correspondences between pairs of isometric surfaces by computing near-rigid correspondences between canonical forms.

Bronstein et al. [6] introduce a representation called *spherical embedding*, which aims to find an embedding in $\mathcal{S} = \mathbb{S}_r^k$, where $\mathbb{S}_r^k$ is a sphere of radius r in k dimensions. Spherical MDS can be computed using a gradient descent approach in $O(tn^2)$ time, where t is the number of iterations needed until convergence. Bronstein et al. use this approach for expression invariant face recognition.

Bronstein et al. [8] introduce *generalized MDS (GMDS)*, which aims to compute a canonical representation in $\mathcal{S} = T$, where T is any triangular manifold mesh. If T contains n vertices, the approach takes $\Omega(tn^2 \log n)$ time, where t is the number of iterations needed until convergence of the gradient. Bronstein et al. [9] use GMDS for face recognition. Furthermore, Bronstein et al. [10] use GMDS to find one-to-one correspondences between pairs of possibly incomplete isometric surfaces.

All of the approaches reviewed so far minimize a global embedding energy. However, in many applications, dissimilarities between similar objects have more importance than those between totally dissimilar objects. Two popular algorithms for dimension reduction that find an embedding in $\mathcal{S} = \mathbb{R}^k$ that minimizes a sum of local embedding energies are *locally linear embedding (LLE)* [29] and *isomap* [33].

3. Bending Invariant Mesh

This section presents an algorithm to compute a bending invariant mesh structure X representing the triangular mesh S. To compute the bending invariant mesh efficiently, a coarse-to-fine strategy is employed. First, we outline how to compute a bending invariant mesh. Second, we derive a coarse-to-fine strategy for improved efficiency.

3.1. Computing a Bending Invariant Mesh

A bending invariant mesh X of a shape S is a deformation of S that has the property that the geodesic distance between each pair of vertices on S is approximated well by the Euclidean distance between the corresponding vertices on X while keeping X intersection-free. To compute the canonical form introduced by Elad and Kimmel [14], an embedding in $\mathbb{R}^k$ is computed such that the Euclidean distances in the embedding approximate the geodesic distances on the mesh well. In our application, $k = 3$. We compute

the geodesic distance $\delta_{i,j}$ between the vertices p_i and p_j for $i, j \in S$ using the fast marching technique introduced by Kimmel and Sethian [20]. Furthermore, we compute confidence values $\omega_{i,j} = 1 - \frac{m^h_{i,j}}{m_{i,j}}$, where $m_{i,j}$ is the number of edges on the geodesic path computed by the fast marching technique from p_i to p_j and where $m^h_{i,j}$ is the number of edges tracing a hole of S on the geodesic path from p_i to p_j. We say that an edge traces a hole of S if the edge crosses a triangle that contains at least one vertex on the hole. We use the geodesic distances $\delta_{i,j}$ as dissimilarities and the confidence values $\omega_{i,j}$ as weights to compute the bending energy of the manifold S. That is, we aim to minimize

$$E_{LS} = \frac{2}{n(n-1)} \sum_{i=1}^{n} \sum_{j=i+1}^{n} \omega_{i,j} \left(\delta_{i,j} - d_{\mathbb{R}^k}(\vec{x}_i, \vec{x}_j)\right)^2 .$$

To maintain a non-self-intersecting mesh during the deformation, we use a repelling energy term. We aim to satisfy $C_{MD} = d(t_i, t_j) > 0$ for all non-adjacent triangles t_i and t_j of X, where $d(t_i, t_j)$ denotes the Euclidean distance between t_i and t_j. The Euclidean distance between t_i and t_j is computed as the minimum of the Euclidean distance between any edge of t_i and any edge of t_j, and the Euclidean distance between any vertex of t_i (respectively t_j) whose perpendicular projection onto the supporting plane of t_j (respectively t_i) is located at the interior of t_j (respectively t_i) and the supporting plane of t_j (respectively t_i). Ladd and Kavraki [23] discuss how to compute the Euclidean distance between two edges using dot products only.

This repelling term is incorporated in the minimization problem by minimizing

$$E = (1-\lambda)E_{LS} + \lambda \frac{1}{m} \sum_{t_i} \frac{1}{\min_{t_j}(d(t_i, t_j)^2)},$$

where m denotes the number of triangles of X, over all non-adjacent triangles t_i and t_j.

The constant $0 \leq \lambda \leq 1$ regulates how much weight is given to each of the two terms of E. We only wish to give weight to the repelling term if two non-adjacent triangles almost intersect each other. This is achieved by choosing λ to be small. When two triangles come close to intersecting each other, the repelling term tends towards infinity. In this case, the repelling term will have significant influence even when multiplied by a small λ. Hence, we set $\lambda = 10^{-10}$ in our experiments.

Note that the gradient of E with respect to all the vertex positions can be computed analytically. The gradient of E_{LS} with respect to the vertex positions is used in the SMACOF algorithm explained by Borg and Groenen [4, p.146-155]. The gradient of the repelling energy term can be computed with the help of the gradient of the Euclidean distance between two edges used by Ladd and Kavraki [23]. As the gradient can be computed analytically, E can be minimized using a gradient descent or a quasi-Newton approach.

We start from the given mesh S and compute the bending invariant mesh X by moving the vertices of S until E is minimized. We use the limited-memory Broyden-Fletcher-Goldfarb-Shanno quasi-Newton approach [25] to minimize E in our implementation.

3.2. Coarse-To-Fine Strategy

To compute the bending invariant mesh efficiently in terms of time and space, a coarse-to-fine strategy is employed. First, we compute uniformly distributed sample sets P containing n' vertices from S using Voronoi sampling [27]. Second, we compute the bending invariant mesh of this sample set. Third, we add the other points to the bending invariant mesh one by one.

To compute the bending invariant mesh of the sample set P, we need a low-resolution mesh that only contains the samples. We compute this reduced mesh S_P using edge collapses. We collapse all the edges until only vertices in P remain. Of all the edges that do not introduce topological changes or self-intersections of the mesh when collapsed, we always collapse the edge that results in the least change in volume. To find this edge, we use a simplified version of the approach by Lindstrom and Turk [24]. When collapsing the edge $e = (v_0, v_1)$, we collapse v_0 into v_1. Hence, every triangle $t = (v_0, v_i, v_j)$ becomes $t' = (v_1, v_i, v_j)$. We can evaluate the change in volume caused by the edge collapse by summing all of the signed volumes of the tetrahedra (v_1, v_0, v_i, v_j). We use this approximation of the volume change even when holes are present in S. Once the mesh S_P is found, we compute the bending invariant mesh X_P as outlined in the previous section.

Next, all vertices of $S \backslash P$ are added to the bending invariant mesh. We initialize the vertices using the mean-value geometry encoding [21]. We then minimize for the vertices $\vec{x}_{n'+j}, j = 1, \ldots, n-n'$, the energy

$$E^* = (1-\lambda)E^*_{LS} + \lambda \frac{1}{m} \sum_{t_i} \frac{1}{\min_{t_j}(d(t_i, t_j)^2)},$$

where

$$E^*_{LS} = \frac{1}{n'(n-n')} \sum_{j=n'+1}^{n} \sum_{i=1}^{n'} \omega_{j,i} \left(\delta_{j,i} - d_{\mathbb{R}^k}(\vec{x}_j, \vec{x}_i)\right)^2 ,$$

where m is the number of triangles in X, and where t_i and t_j are non-adjacent triangles of X. The gradient of E^* with respect to $\vec{x}_{n'+j}$ can be computed explicitly and we use the same minimization scheme as above. This results in the bending invariant mesh X.

3.3. Analysis

To compute a bending invariant mesh X, we first find a sample set P of size n' using Voronoi sampling in $O(n'n \log n)$ time.

Second, we compute a low-resolution mesh S_P using edge collapses. During the course of the algorithm, there are $O(n)$ edges that are candidates for an edge collapse. Computing the associated volume change for each of these edges takes $O(1)$ time on average, since this computation only depends on the neighborhood of the edge. Each time an edge is collapsed, we ensure that collapsing the edge does not cause self-intersections of the mesh. This step takes $O(n)$ time per edge. Hence, computing S_P takes $O(n^2)$ time.

To improve the efficiency of this step in our implementation, we only test self-intersections with a set of triangles found using nearest neighbor searches on the vertices of S. For each vertex v of a new triangle t created by an edge collapse, we find the $k = deg(v) + 1$ nearest neighbors in the current partially compressed version of S, where $deg(v)$ denotes the degree of v after the collapse. We then insert all of the triangles that are incident to one of the k nearest neighbors of v and that are not adjacent to t into the set of triangles used to test self-intersections. We find the k nearest neighbors of v with the Approximate Nearest Neighbor library [1], [2]. Although this approach eliminates the guarantee that no self-intersections occur, we did not encounter any problems in our experiments.

Third, we compute X_P by minimizing E. Evaluating E and its gradient with respect to all the vertex positions takes $O(n'^2)$ time. We improve the efficiency of computing the term $\sum_{t_i} \frac{1}{\min_{t_j}(d(t_i,t_j)^2)}$ in our implementation with the same technique used when computing S_P.

The quasi-Newton approach evaluates E and its gradient in each iteration. Hence, computing X_P takes $O(un'^2)$ time, where u is the number of iterations.

Finally, we add all vertices of $S \backslash P$ to the bending invariant mesh. This takes $O(n)$ time. Moving each of the added points by minimizing E^* takes $O(un^2)$ time in the worst case. As before, we improve the efficiency of computing $\sum_{t_i} \frac{1}{\min_{t_j}(d(t_i,t_j)^2)}$ in our implementation using a kd-tree.

The analysis of all the steps of the algorithm shows that computing the bending invariant mesh X takes $O(un^2)$ time. Furthermore, the algorithm uses $O(n + n'^2)$ space.

3.4. Examples

We compute the bending invariant meshes for a set of four poses of an Alien model derived from the Princeton shape benchmark [31] on an Intel Pentium D with 3.5 GB of RAM. We chose a model of an alien from the Princeton Shape Benchmark and animated the model to obtain multiple postures with known correspondences using the automatic technique by Baran and Popović [3]. The models contain 429 vertices each.

Figure 1 shows the bending invariant meshes. The back faces of the models are colored blue. The first row shows the models. The second row shows the canonical forms of the meshes computed using E_{LS}. We can see that, especially in the leftmost model, many self-intersections occur. The third row shows the bending invariant meshes computed using E. The computation of these models took about 2 minutes on average. The fourth row shows the bending invariant meshes computed using the coarse-to-fine strategy when $n' = 250$. The computation of these models also took about 2 minutes on average. For the results in rows three and four, few self intersections occur. The meshes in row four are slightly noisier than the meshes in row three because only 250 samples are used to minimize E. Otherwise, the results obtained using full resolution and the results obtained using a coarse-to-fine strategy are similar.

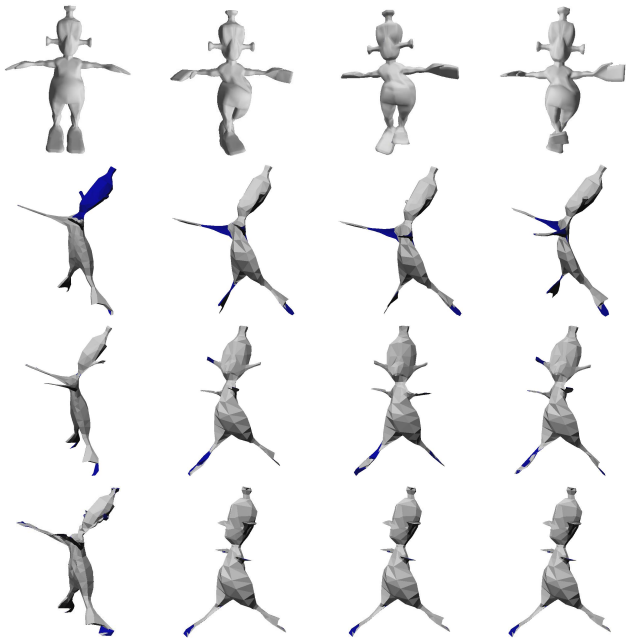


Figure 1. *Four poses of an alien model and their canonical forms. The first row shows the models. The second row shows the canonical forms of the meshes computed using E_{LS}. The third row shows the bending invariant meshes computed using E. The fourth row shows the bending invariant meshes computed using the coarse-to-fine strategy.*

4. Application to Groupwise Correspondence

We apply bending invariant meshes to find dense point-to-point correspondences between s deformed surfaces $S^{(1)}, S^{(2)}, \ldots, S^{(s)}$ corresponding to different postures of the same non-rigid object.

First, we compute the bending invariant meshes $X^{(1)}, X^{(2)}, \ldots, X^{(s)}$. Second, we present an iterative ap-

proach to compute the correspondences between the near-rigid meshes $X^{(r)}$. This approach uses the average shape $\overline{X}$ of the corresponded points of the meshes $X^{(r)}$.

The iterative approach proceeds in three steps. First, the alignment of the bending invariant meshes is initialized. Furthermore, the average shape is initialized. For this step, surface-based feature points on X are used. The initial alignment is an important step in this algorithm because the subsequent iteration uses nearest neighbor registration and can therefore easily get trapped in local minima. Section 4.1 discusses how to find the initial alignment.

Second, we update the current correspondence between $X^{(r)}$ and the average shape. This step uses levels of nearest neighbors and is outlined in Section 4.2.

Third, we find the best rotational alignment of the meshes $X^{(r)}$ via generalized Procrustes analysis (GPA). In this step, we assume that a correspondence between $X^{(r)}$ is given. Section 4.3 discusses this step.

Note that the third step yields a new average shape, which is used in the following iteration. We iterate steps two and three of the algorithm. We can evaluate the quality of the correspondence by summing the Euclidean distances between corresponding vertices over all meshes. The iteration stops once the quality of the correspondence improved by less than $\epsilon^{(corres)}$. We also stop the iteration after a maximum number $t^{(max)}$ of iterative steps. In our experiments, we set $\epsilon^{(corres)} = 10^{-5}$ and $t^{(max)} \in [10, 100]$.

4.1. Initializing the Alignment

We start by rigidly aligning the meshes $X^{(r)}$ such that corresponding points are close to each other. This step of the algorithm is important because the subsequent iteration uses nearest neighbor registration and can therefore easily get trapped in local minima. Finding the best rigid alignment between two bending invariant representations based on vertex coordinates only is a hard problem [19], [34]. Since $X^{(r)}$ is an intersection-free mesh, we use mesh-based features on $X^{(r)}$ to find the initial alignment.

Let $n^{(min)} = \min(n^{(r)})$ denote the minimum number vertex count over all the meshes $X^{(r)}$ and let $n^{(max)} = \max(n^{(r)})$ denote the maximum number vertex count over all the meshes $X^{(r)}$. Let $X^{(min)}$ denote the mesh $X^{(r)}$ that consists of $n^{(min)}$ vertices. If there is more than one mesh $X^{(r)}$ that consists of $n^{(min)}$ vertices, choose $X^{(min)}$ as the one that yields the best quality of the correspondence.

We start by initializing the alignment of the meshes as follows. We rotate each mesh $X^{(r)}$, such that $X^{(r)}$ is aligned with the principal axes defined by $X^{(r)}$'s vertices. Note that four different rotations can achieve this alignment. We fix the alignment of $X^{(min)}$ by choosing one of the four rotations arbitrarily. We initialize $\overline{X} = X^{(min)}$.

For each mesh $X^{(r)} \neq X^{(min)}$, we choose of the four possible rotations the one that minimizes a mesh-based feature matching cost between $X^{(r)}$ and $X^{(min)}$.

To find and match the features, we use a simplification of the algorithm introduced by Gelfand et al. [16]. To start, we compute the mean curvature on each of the bending invariant meshes using the approach by Meyer et al. [26]. We use the mean curvature as this measure is closely related to the integral volume descriptor developed by Gelfand et al.

To extract the features of $X^{(r)}$, we find the points that have the most unusual mean curvature by computing a histogram of the mean curvature values according to Scott's rule [30]. That is, we compute a histogram with bin width $b = 3.49\sigma n^{(min)-\frac{1}{3}}$, where σ is the standard deviation of the mean curvature values. To select feature points, we find the least populated bins containing a total of at most $0.01n^{(min)}$ points. We do not select feature points that are in the 1-ring neighborhood of an already selected feature point. This approach yields $k^{(r)}$ feature points on $X^{(r)}$.

For each ordered pair of meshes $X^{(r_1)}$ and $X^{(r_2)}$, we find $k^{(r_2)}$ points on $X^{(r_1)}$ that correspond to the $k^{(r_2)}$ feature points on $X^{(r_2)}$ as follows. For each of the $k^{(r_2)}$ feature points on $X^{(r_2)}$, we find the point on $X^{(r_1)}$ that is closest to the feature point in terms of mean curvature. This is the matching feature point. This way, we obtain $\sum_{r=1}^{s} k^{(r)}$ matching feature points on each mesh $X^{(r)}$.

Once the matching features are found, we test for $X^{(r)}$ all four rotational alignments that align $X^{(r)}$ with the principal axes defined by its vertices. For each of the alignments, we compute as cost the sum of squared Euclidean distances between the matched feature points on $X^{(r)}$ and $X^{(min)}$. Finally, we choose the alignment that yields the minimum cost.

4.2. Updating the Correspondence

This section explains how to update the correspondence between a group of meshes $X^{(r)}$ and their average $\overline{X}$. The aim is to register each mesh $X^{(r)}$ to the shape $\overline{X}$. We perform this correspondence using closest pairs of vertices.

In a first step, we find for each vertex p in $X^{(r)}$ its nearest neighbor $n(p)$ in $\overline{X}$. If p is the nearest neighbor of $n(p)$ in $X^{(r)}$, then $n(p)$ is the vertex of $\overline{X}$ corresponding to p and for the following steps, both p and $n(p)$ are removed from consideration. In the k-th step, we find for each vertex p in $X^{(r)}$ that is left for consideration its nearest neighbor $n(p)$ in the part of $\overline{X}$ that is left for consideration. As before, if p is the nearest neighbor of $n(p)$ in the part that is left for consideration of $X^{(r)}$, then the two vertices are corresponded and for the following steps, both p and $n(p)$ are removed from consideration. We use kd-trees to efficiently compute pairs of nearest neighbors.

Once the correspondence is updated, exactly $n^{(min)}$ vertices of $X^{(r)}$ have a corresponding vertex in $\overline{X}$. We sort these vertices in the same order as the vertices in $\overline{X}$ and we denote this ordered set by $O^{(r)}$ in the following.

4.3. Computing the Best Alignment via GPA

Given the set $O^{(r)}$ of s shapes with corresponding vertices, we perform generalized Procrustes analysis (GPA) as outlined in Dryden and Mardia [13]. Each shape $O^{(r)}$ is rotated in turn to optimally fit the average of the shapes $O^{(1)}, \ldots, O^{(r-1)}, O^{(r+1)}, \ldots, O^{(s)}$. While we only use the vertices of $O^{(r)}$ to compute the optimal rotations, we rotate the full meshes $X^{(r)}$. These steps are iterated until the quality of the correspondence improved by less than $\epsilon^{(corres)}$ or until $t^{(max)}$ iterative steps have been executed. Once the optimal alignment of the shapes is found, the average $\overline{X}$ is recomputed as the average of the shapes $O^{(r)}$.

4.4. Analysis

To compute the groupwise correspondence between s given meshes, we first find the bending invariant meshes in $O(sn^{(max)2})$ time as outlined in Section 3.

To find the initial alignment of the bending invariant meshes, we first find $k^{(r)}$ feature points on $X^{(r)}$ in $O(k^{(r)}n^{(r)})$ time. Second, we find the matching feature points for each ordered pair of meshes in $O(s^2k^{(max)}n^{(max)})$ time, where $k^{(max)}$ is the maximum of all $k^{(r)}$. Finally, for each $X^{(r)} \neq X^{(min)}$, we find the best of four possible alignments in $O(s^2k^{(max)})$ time. Hence, the total time of finding the initial alignment of the meshes is $O(s^2k^{(max)}n^{(max)})$.

To update the correspondence between the bending invariant meshes and their current average $\overline{X}$, we use a kd-tree on the vertices of $\overline{X}$. Building the kd-tree takes $O(n^{(min)} \log n^{(min)})$ time. Next, we find for each vertex p in $X^{(r)}$ its nearest neighbor $n(p)$ in $\overline{X}$. This takes $O(sn^{(max)}(n^{(min)})^{2/3})$ time. We repeat these steps until all points on $\overline{X}$ have a corresponding point in $X^{(r)}$. In the worst case, we need to repeat $n^{(min)}$ times. Hence, the worst-case running time of this step is $O(sn^{(max)}n^{(min)}(n^{(min)})^{2/3})$. Note however that in practice, we expect to repeat less often than $n^{(min)}$ times.

To compute the best alignment via GPA, we solve a linear system of equations via least-squares fitting. This takes $O(n^{(min)2})$ time. Since there are s shapes that need to be aligned and since we repeat the alignment u_{GPA} times, the total time of this step is $O(su_{GPA}n^{(min)2})$.

To find the groupwise correspondence, u iterations are performed. This takes $O(u(sn^{(max)}n^{(min)}(n^{(min)})^{2/3} + su_{GPA}n^{(min)2}))$ time.

5. Experimental Results

The experiments were conducted using an implementation in C++ using OpenMP on an Intel Pentium D with 3.5 GB of RAM.

Figure 2(a) shows the correspondences that are computed for the four alien models when 250 samples are used to compute bending invariant meshes. Corresponding vertices are shown using the same color. For this experiment $t^{(max)} = 100$. Recall that we know the ground truth correspondences for this model. We can therefore evaluate the accuracy of the computed correspondences.

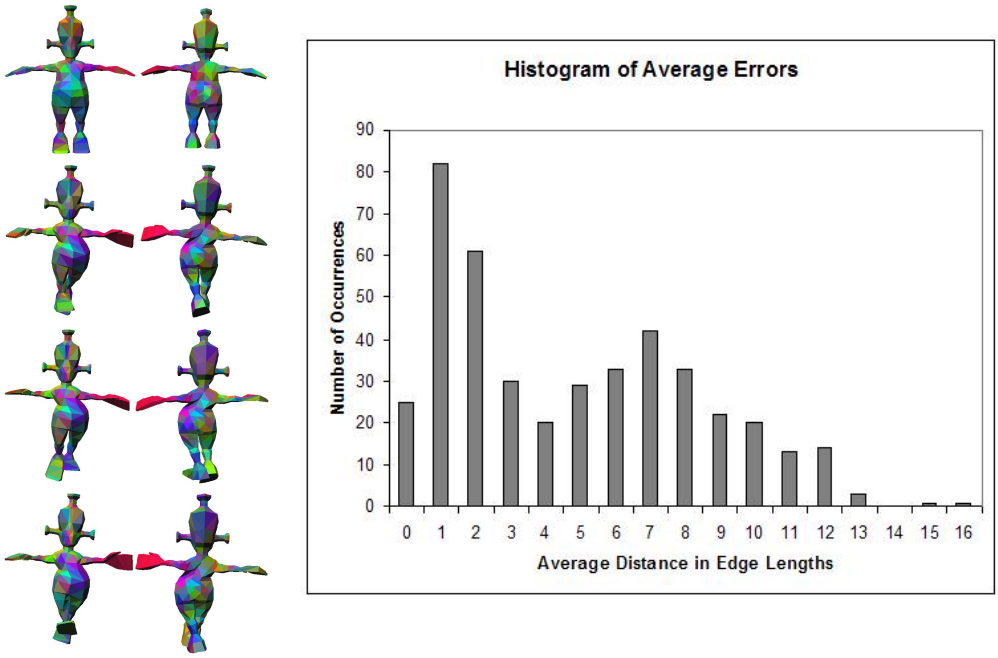
Let $p_k, k = 0, \ldots, n^{(min)}$ denote the vertices of $\overline{X}$. Let $p_k^{(r)}, k = 0, \ldots, n^{(r)}$ denote the vertices of $X^{(r)}$. Let the vertices of the meshes $X^{(r)}$ be ordered, such that the true correspondence of $p_k^{(i)}$ is $p_k^{(j)}$, for i, j from 1 until s. Furthermore, let $a^{(r)}$ be a function, such that $p^{(r)}_{a^{(r)}(k)}$ is the correspondence of p_k in $S^{(r)}$ computed by our algorithm. We compute the *average correspondence error* as

$$\widehat{E}_{COR} = \frac{\sum_{i=1}^{s}\sum_{j=1}^{s} d_{S^{(i)}}\left(p^{(i)}_{a^{(i)}(k)}, p^{(i)}_{a^{(j)}(k)}\right)}{s(s-1)},$$

where $d_{S^{(i)}}\left(p^{(i)}_{a^{(i)}(k)}, p^{(i)}_{a^{(j)}(k)}\right)$ denotes the number of edges on the Dijkstra path from $p^{(i)}_{a^{(i)}(k)}$ to $p^{(i)}_{a^{(j)}(k)}$ along $S^{(i)}$. Figure 2(b) shows a histogram of the average correspondence errors $\widehat{E}_{COR}$. We can see that most correspondences are accurate within two edge lengths.

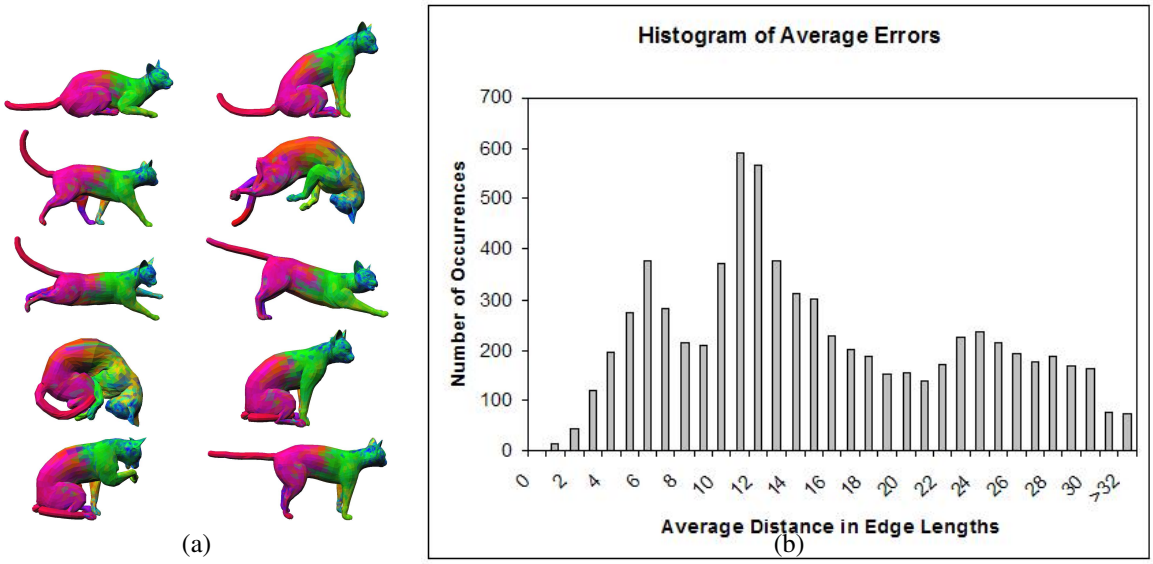
Furthermore, we evaluate the approach using a set of ten cat models. The models contain 7207 vertices and were created and used by Sumner et al. [32]. We use 1000 samples to compute the bending invariant meshes. On average, this step takes about 11.7 minutes per cat. To improve the efficiency of this experiment, we set $t^{(max)} = 10$. Figure 3 (a) shows the result. Corresponding vertices are shown using the same color. For this set of models the ground truth correspondence is known. Figure 3 (b) shows a histogram of the average correspondence errors $\widehat{E}_{COR}$. We can see that most correspondences are accurate within 13 edge lengths. The error for this experiment is larger than for the previous experiment because the bending invariant meshes are not entirely rigid to each other for the cat models. The accuracy of the correspondence could be improved by allowing for non-rigid registration of the bending invariant meshes. We leave this for future work.

Finally, we demonstrate that the accuracy of the correspondence is strongly related to the amount of non-isometric deformation. We use the set of four alien models with small deformation shown in Figure 4 that were derived from the Princeton shape benchmark [31] as above. We compute the groupwise correspondence between the four alien models. Instead of computing $\widehat{E}_{COR}$ over the set of four models, we compute the correspondence error between two given poses $S^{(i)}$ and $S^{(j)}$ as $\widehat{E}^{(i,j)}_{COR} = d_{S^{(i)}}\left(p^{(i)}_{a^{(i)}(k)}, p^{(i)}_{a^{(j)}(k)}\right)$. Furthermore, for each edge e, we compute the amount of non-isometric deformation $D^{(i,j)}$ as the difference in length of e in poses $S^{(i)}$ and $S^{(j)}$. We use $D^{(i,j)}$ although it depends on the scale of the mod-



(a) (b)

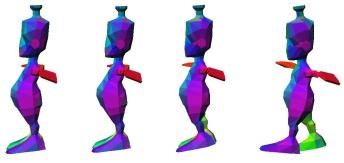
Figure 2. *(a): Front and back views of the correspondence between four poses of an alien model. (b): Histogram of* $\widehat{E}_{COR}$.



(a) (b)

Figure 3. *(a): Correspondence between ten poses of a cat model. (b): Histogram of* $\widehat{E}_{COR}$.

els because all four models have the same scale. Table 1 shows the average and maximum $\widehat{E}^{(i,j)}_{COR}$ and the average and maximum $D^{(i,j)}$ for three pairs of aliens. Note that as expected, there is a direct correlation between the amount of non-isometric deformation and correspondence error.



$S^{(1)}$ $S^{(2)}$ $S^{(3)}$ $S^{(4)}$

Figure 4. *Correspondence between four alien models.*

6. Conclusion

This paper introduces a new bending invariant representation of a possibly incomplete triangular mesh S. The representation preserves the intersection-free mesh structure of

Models	$S^{(1)}, S^{(2)}$	$S^{(1)}, S^{(3)}$	$S^{(1)}, S^{(4)}$
Average $D^{(i,j)}$	$7.7 \cdot 10^{-5}$	$4.9 \cdot 10^{-4}$	$8.3 \cdot 10^{-4}$
Max $D^{(i,j)}$	$2.0 \cdot 10^{-3}$	$9.0 \cdot 10^{-3}$	$1.7 \cdot 10^{-2}$
Average $\widehat{E}_{COR}^{(i,j)}$	0.45	0.93	1.96
Max $\widehat{E}_{COR}^{(i,j)}$	4	14	12

Table 1. *Table demonstrates the correlation between $D^{(i,j)}$ and $\widehat{E}_{COR}^{(i,j)}$. As the amount of non-isometric deformation increases (left to right in table), the correspondence error also increases.*

S and can therefore be used to compute bending invariant feature points based on local surface properties. We apply the new representation to solve the groupwise point-to-point correspondence problem. In this application, we use features derived from the mean curvature on the bending invariant representation.

We leave the following ideas for future work. First, by using features that are less sensitive with respect to noise, we can improve the initial alignment of the shapes. Second, by allowing for non-rigid alignments of the bending invariant meshes, we can improve the accuracy of the correspondence. Third, by using different local surface properties on the bending invariant mesh, we can extract feature points that are bending invariant. These bending invariant features can potentially be used for various applications.

References

[1] S. Arya and D. M. Mount. Algorithms for fast vector quantization. In *DCC*, pages 381–390, 1993.

[2] S. Arya and D. M. Mount. Approximate nearest neighbor queries in fixed dimensions. In *SODA*, pages 271–280, 1993.

[3] I. Baran and J. Popović. Automatic rigging and animation of 3D characters. *ACM TOG*, 26(3), 2007.

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

[5] A. Bronstein, M. Bronstein, and R. Kimmel. *Numerical Geometry of Non-Rigid Shapes*. Springer, 2008.

[6] A. M. Bronstein, M. M. Bronstein, and R. Kimmel. Expression-invariant face recognition via spherical embedding. In *IEEE ICIP*, 2005.

[7] A. M. Bronstein, M. M. Bronstein, and R. Kimmel. Three-dimensional face recognition. *IJCV*, 64(1):5–30, 2005.

[8] A. M. Bronstein, M. M. Bronstein, and R. Kimmel. Generalized multidimensional scaling: a framework for isometry-invariant partial surface matching. *PNAS*, 103(5):1168–1172, 2006.

[9] A. M. Bronstein, M. M. Bronstein, and R. Kimmel. Robust expression-invariant face recognition from partially missing data. In *ECCV*, pages 396–408, 2006.

[10] A. M. Bronstein, M. M. Bronstein, and R. Kimmel. Calculus of non-rigid surfaces for geometry and texture manipulation. *IEEE Trans. Vis. Comp. Graphics*, 13(5):902–913, 2007.

[11] A. M. Bronstein, M. M. Bronstein, and R. Kimmel. Topology-invariant similarity of nonrigid shapes. *IJCV*, To appear.

[12] A. M. Bronstein, M. M. Bronstein, R. Kimmel, and I. Yavneh. Multigrid multidimensional scaling. *NLAA*, 13(2–3):149–171, 2006.

[13] I. Dryden and K. Mardia. *Statistical Shape Analysis*. Wiley, 2002.

[14] A. Elad and R. Kimmel. On bending invariant signatures for surfaces. *IEEE PAMI*, 25(10):1285–1295, 2003.

[15] C. Faloutsos and K.-I. Lin. Fastmap: A fast algorithm for indexing, data-mining and visualization of traditional and multimedia datasets. In *ACM SIGMOD*, 1995.

[16] N. Gelfand, N. J. Mitra, L. J. Guibas, and H. Pottmann. Robust global registration. In *SGP*, page 197, 2005.

[17] J. C. Gower. Some distance properties of latent root and vector methods used in multivariate analysis. *Biometrika*, 53:325–338, 1966.

[18] V. Jain and H. Zhang. A spectral approach to shape-based retrieval of articulated 3d models. *CAD*, 39(5):398–407, 2007.

[19] V. Jain, H. Zhang, and O. van Kaick. Non-rigid spectral correspondence of triangle meshes. *IJSM*, 13(1):101–124, 2007.

[20] R. Kimmel and J. Sethian. Computing geodesic paths on manifolds. *PNAS*, 95:8431–8435, 1998.

[21] V. Kraevoy, A. Sheffer, and C. Gotsman. Mean-value geometry encoding. *IJSM*, 12(1):29–46, 2006.

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

[23] A. Ladd and L. Kavraki. Using motion planning for knot untangling. *The International Journal of Robotics Research*, 23(7–8):797–808, 2004.

[24] P. Lindstrom and G. Turk. Fast and memory efficient polygonal simplification. In *IEEE Vis.*, pages 279–286, 1998.

[25] D. C. Liu and J. Nocedal. On the limited memory method for large scale optimization. *Math. Prog.*, 45:503–528, 1989.

[26] M. Meyer, M. Desbrun, P. Schröder, and A. Barr. *Discrete Differential-Geometry Operators for Triangulated 2-Manifolds*. In: Hege, Polthier. Visualization and Mathematics III. Chapter 2:35–58. Springer, 2003.

[27] C. Moenning and N. Dodgson. Fast marching farthest point sampling. In *EUROGRAPHICS*, 2003.

[28] G. Rosman, A. M. Bronstein, M. M. Bronstein, and R. Kimmel. Topologically constrained isometric embedding. In *ICMLPR*, 2006.

[29] S. Roweis and L. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 190(5500):2323–2326, 2000.

[30] D. W. Scott. On optimal and data-based histograms. *Biometrika*, 66(3):605–610, 1979.

[31] P. Shilane, P. Min, M. Kazhdan, and T. Funkhouser. The Princeton shape benchmark. In *SMI*, 2004.

[32] R. W. Sumner and J. Popović. Deformation transfer for triangle meshes. *ACM TOG*, 23(3):399–405, 2004.

[33] J. B. Tenenbaum, V. de Silva, and J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 190(5500):2319–2323, 2000.

[34] S. Wuhrer, C. Shu, Z. B. Azouz, and P. Bose. Posture invariant correspondence of incomplete triangular manifolds. *IJSM*, 13(2):139–157, 2007.