International Journal of Shape Modeling

Posture Invariant Correspondence of Incomplete Triangular Manifolds

Stefanie Wuhrer
School of Computer Science, Carleton University, Ottawa, Ontario, Canada
National Research Council of Canada, Ottawa, Ontario, Canada
swuhrer@scs.carleton.ca

Chang Shu
National Research Council of Canada, Ottawa, Ontario, Canada
chang.shu@nrc-cnrc.gc.ca

Prosenjit Bose
School of Computer Science, Carleton University, Ottawa, Ontario, Canada
jit@scs.carleton.ca

Zouhour Ben Azouz
National Research Council of Canada, Ottawa, Ontario, Canada
zouhour.benazouz@nrc-cnrc.gc.ca

Received (Day Month Year)
Revised (Day Month Year)
Accepted (Day Month Year)

Communicated by (xxxxxxxxxx)

We present an approach to find dense point-to-point correspondences between two deformed surfaces corresponding to different postures of the same non-rigid object in a fully automatic way. The approach requires no prior knowledge about the shapes being registered or the initial alignment of the shapes. We consider surfaces that are represented by possibly incomplete triangular meshes. We model the deformations of an object as isometries. To solve the correspondence problem, our approach maps the intrinsic geometries of the surfaces into a low-dimensional Euclidean space via multi-dimensional scaling. This results in posture-invariant shapes that can be registered using rigid correspondence algorithms.

1. Introduction

We consider the problem of finding *dense point-to-point correspondences* between two deformed surfaces $S^{(1)}$ and $S^{(2)}$ corresponding to different postures of the same non-rigid object. That is, given a position $x^{(1)}$ on $S^{(1)}$, we aim to find the position $x^{(2)}$ on $S^{(2)}$ that corresponds to the same intrinsic location on $S^{(2)}$ as does $x^{(1)}$ on $S^{(1)}$. If the position $x^{(2)}$ is absent on $S^{(2)}$ due to incomplete data, no correspondence is found for $x^{(1)}$. Finding dense point-to-point correspondences

between two deformed surfaces is a key problem in various applications such as mesh deformation and animation [3], shape registration [21], object recognition [17], and mesh parameterization [20], [27]. The 3D models used in these applications usually come from digitizing real-world objects from a discrete set of measurements using a 3D laser-range scanner or image-based reconstruction. Therefore, the reconstructed surfaces are often incomplete.

The main difficulty in finding point-to-point correspondences is that local regions on the surface are often not distinctive. Hence, finding the correspondence for all object points corresponds to searching a large set of candidate correspondences. Previous methods to find point-to-point correspondences for deformable surfaces either restrict the search space using prior knowledge about the objects being registered [1] or use probabilistic methods to solve the problems [2] which has the drawback of producing inaccurate correspondences.

The method proposed in this paper considers surfaces that are represented by possibly incomplete triangular meshes. We model deformations of an object as isometries. We aim to find dense point-to-point correspondences between two incomplete triangular surfaces $S^{(1)}$ and $S^{(2)}$ consisting of $n^{(1)}$ and $n^{(2)}$ vertices, respectively, using a fully automatic correspondence algorithm that does not assume knowledge about markers or template shapes.

We propose an approach to solve the registration problem that can be viewed as an extension of Jain et al.'s approach [18] and that is related to Bronstein et al.'s approach [7]. Both the approach by Jain et al. and the approach by Bronstein et al. have only been used to compute the correspondence between a small set of samples consisting of no more than 3000 vertices. This is due to the high complexity of the algorithms. We extend the approach of Jain et al. by applying a course-to-fine strategy, thereby making the approach useful for real-life data sets containing tens of thousands of vertices.

Our approach consists of three main steps. First, we compute uniformly distributed sample sets $P^{(r)}$ containing $n'^{(r)}$ vertices from $S^{(r)}$ for $r = 1, 2$. We compute the geodesic distance $\delta_{i,j}$ between the vertices p_i and p_j for $i, j \in P^{(r)}$ using the fast marching technique introduced by Kimmel and Sethian [19]. 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 $S^{(r)}$ on the geodesic path from p_i to p_j. We say that an edge traces a hole of $S^{(r)}$ 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 embed the samples $P^{(r)}$ of the manifold $S^{(r)}$ via multi-dimensional scaling (MDS). Let $X^{(r)}$ denote the embedding. The approach embeds $P^{(r)}$ into a low-dimensional Euclidean space $\mathbb{R}^k$ using least-squares MDS.

Second, we compute the rigid correspondence between the posture-invariant canonical forms. Our approach employs the Hungarian method [24] for this compu-

tation. The Hungarian method finds a maximum weight matching in a weighted bipartite graph.

Third, all vertices of $S^{(r)} \setminus P^{(r)}$ are projected to the embedding space. The approach finds the correspondence of the projected vertices by evaluating an approximating thin-plate spline mapping the embedding $X^{(1)}$ to $X^{(2)}$.

The paper is organized as follows. Section 2 reviews previous work on the registration problem. Section 3 gives an overview of the algorithm. Sections 4 and 5 describe and analyze the approach taken in this paper. Section 6 shows experimental results and discusses limitations of the approach. Finally, Section 7 concludes and gives ideas for future work. Our approach is compared to the approaches taken by Jain et al. [18] and Bronstein et al. [7].

2. Related Work

This section reviews previous work aiming to find dense point-to-point correspondences between two deformed surfaces corresponding to different postures.

Allen et al. [1] deform a known template mesh of a generic human body to fit a range scan of a human body. The deformation is guided by a small set of known marker positions on the object and it is ensured that the deformation is smooth in the neighborhood of each vertex. While this method performs well, a template model as well as a sparse set of marker positions need to be available.

Recently, markerless registration methods have received considerable attention [2], [18], [7]. Anguelov et al. [2] model the registration problem using a Markov network. Two surfaces are registered by maximizing a joint probabilistic model over all correspondences. The method aims to preserve geodesic distances on the surface. Furthermore, the method ensures that close by points in one surface map to close by points in the other surface. This method encounters problems when registering surfaces of a human body due to symmetry alignments.

Jain et al. [18] and Bronstein et al. [7] solve the non-rigid correspondence problem by embedding the intrinsic geometry of the surface into a suitable embedding space using MDS. Denote the two surfaces being registered by S and Q. Jain et al. embed the intrinsic geometries of both S and Q into Euclidean space using classical MDS [14]. MDS aims to embed the points on S and Q, respectively, into Euclidean space $\mathbb{R}^k$, such that the geodesic distances on S and Q, respectively, are approximated well by the Euclidean distances between the corresponding embedded points. The embedded surfaces are then aligned in embedding space, which yields a one-to-one correspondence between S and Q. This fully automatic approach does not assume a template mesh or marker positions to be known. The approach was shown to perform well for surfaces with intrinsic geometry that can be represented well in Euclidean spaces. For surfaces with non-Euclidean intrinsic geometry, symmetry alignment problems may arise. The main drawback of this approach is its quadratic time and space complexity that restricts the use of this method to small models. The experiments of Jain et al. compute correspondences for models containing between

180 and 250 vertices.

We extend this approach and make it applicable to real-life data sets by implementing a coarse-to-fine strategy. Another drawback of Jain et al.'s approach is that the surface models can only contain insignificant amounts of missing data. We overcome this problem by using weighted geodesic distances during the embedding. The embedding is then performed using least-squares MDS [5] . This allows us to register surfaces with significant amounts of missing data.

Bronstein et al. embed the intrinsic geometry of S into the surface Q using generalized MDS [6]. The approach also employs a coarse-to-fine strategy to overcome the high time and space complexity of the embedding problem. Generalized MDS aims to embed the points on S into the surface Q, such that the geodesic distance on S is approximated well by the geodesic distance of the corresponding points on Q. This method avoids the large embedding errors caused by embedding into Euclidean space. Bonstein et al. show that the method performs well in practice. The method was tested on data sets of up to 3000 vertices obtained by coarsening larger data sets [28].

Bronstein et al. point out that the method is suitable for registration of incomplete surfaces. However, note that this claim is correct only if one of the surfaces to be registered is complete. If S is an incomplete surface and Q is a complete surface, the geodesic distances on S can be weighted similarly to our approach. However, consider the registration of two incomplete surfaces S and Q. In order to embed the intrinsic geometry of S into the surface Q, the GMDS algorithm repeatedly computes geodesic distances on Q. These distances are taken to be accurate by the GMDS algorithm. If the surface Q is incomplete, this results in large embedding errors. Hence, to use the algorithm by Bronstein et al., at least one of the surfaces to be registered needs to be almost complete to avoid large embedding errors. When the aim is to register two incomplete surfaces, a template mesh is required. However, the approach by Bronstein et al. does not require prior knowledge about marker positions.

3. Correspondence via least-squares multi-dimensional scaling

We find the point-to-point correspondence between vertices on $S^{(1)}$ and $S^{(2)}$, respectively, using a method that extends the work of Jain et al. [18]. One limitation of Jain et al.'s approach is its quadratic space complexity. This complexity stems from the computation of the canonical form and limits the scalability of the approach. The models considered for Jain et al.'s experiments all have at most 250 vertices. We overcome this limitation and make the approach applicable to real-world data consisting of tens of thousands of vertices by taking a coarse-to-fine approach.

Our algorithm is summarized in Algorithm 1 and outlined in more detail in Section 4. The input of the algorithm consists of two possibly incomplete triangular manifolds $S^{(1)}$ and $S^{(2)}$ consisting of $n^{(1)}$ and $n^{(2)}$ vertices, respectively.

Algorithm 1: *Overview of the Algorithm.*

/* Step 1: Compute samples: */
1 Compute sample sets $P^{(r)}, r = 1, 2$ of size $n'^{(r)}$ on $S^{(r)}$ using Voronoi sampling;
/* Step 2: Compute the correspondence between the two sample sets (coarse correspondence): Lines 2-10 */
2 **for** *Each* p_i, $i \in P^{(r)}$ **do**
3 **for** *Each* $i, j \in P^{(r)}$ **do**
4 Compute the geodesic distance $\delta_{i,j}$ between the vertices p_i and p_j on $S^{(r)}$ using the fast marching technique;
5 Compute confidence values $\omega_{i,j}$ approximating the fraction of the length of the geodesic path between p_i and p_j that does not trace a hole of $S^{(r)}$;
6 **end**
7 **end**
8 Use $\delta_{i,j}$ and $\omega_{i,j}$ to embed the sample vertices $P^{(r)} = \{p_1^{(r)}, \ldots, p_{n'^{(r)}}^{(r)}\}$ into $\mathbb{R}^k$ for a constant embedding dimension k. The embedding is given as $X^{(r)} = \left[X_1^{(r)}, \ldots, X_{n'^{(r)}}^{(r)}\right]$;
9 Find an initial alignment of $X^{(1)}$ and $X^{(2)}$ using eigenmode sign assignments;
10 Compute the correspondence between $X^{(1)}$ and $X^{(2)}$ based on Euclidean distances using the Hungarian algorithm;
/* Step 3: Compute thin-plate spline: */
11 Compute an approximate thin-plate spline matching $\Phi(X^{(1)})$ from $X^{(1)}$ to $X^{(2)}$;
/* Step 4: Compute the correspondence between the non-matched vertices (fine correspondence): Lines 12-19 */
12 **for** *Each* $p \in S^{(r)} \setminus P^{(r)}$ **do**
13 Add p to the embedding $X^{(r)}$;
14 **end**
/* The added points are $X_{n'^{(r)}+1}^{(r)}, \ldots, X_{n^{(r)}}^{(r)}$. */
15 **for** $i = n'^{(1)} + 1, \ldots, n^{(1)}$ **do**
16 Evaluate $\tilde{X}_i^{(1)} = \Phi(X_i^{(1)})$;
17 Find the nearest neighbor $X_j^{(2)}$ of $\tilde{X}_i^{(1)}$ in $X_j^{(2)}, j = n'^{(2)} + 1, \ldots, n^{(2)}$;
18 Choose $X_j^{(2)}$ as corresponding point of $X_i^{(1)}$ if $\tilde{X}_i^{(1)}$ and $X_j^{(2)}$ are nearest neighbors;
19 **end**

4. Description of the Algorithm

In this section, we describe each step of the approach in detail. First, the approach computes a small number of uniformly distributed sample points $P^{(1)}$ and $P^{(2)}$

on $S^{(1)}$ and $S^{(2)}$, respectively, using Voronoi sampling. Uniformity in this context means that the sample density is approximately constant over the surface. Voronoi sampling, also called Farthest Point Sampling, allows to obtain uniformly distributed samples in an iterative way. Farthest point sampling was introduced to image processing by Eldar et al. [13]. It starts from a random sample and iteratively computes the next sample as the vertex which is farthest from the samples computed so far. Combining the fast marching technique and farthest point sampling to obtain samples used to perform MDS works well in practice [12], [18]. Hence, we use farthest point sampling to obtain $P^{(r)}$ and compute an MDS embedding.

Second, we establish a *coarse correspondence* between the sample points $P^{(1)}$ and $P^{(2)}$ as described in Section 4.1. Third, we establish the the *fine correspondence*, that is, the correspondences between vertices in $S^{(1)} \setminus P^{(1)}$ and $S^{(2)} \setminus P^{(2)}$. This step is described in Section 4.2.

4.1. *Coarse Correspondence*

We establish the coarse correspondence, which is the correspondence between the two sample sets, using a variation of the approach introduced by Jain et al. [18]. For the sample set $P^{(r)}, r = 1, 2$, we compute the pairwise geodesic distances $\delta_{i,j}$ on $S^{(r)}$ between the vertices p_i and p_j for $i, j \in P^{(r)}$ using the fast marching technique [19]. In fact, due to the close relationship between the fast marching technique and farthest point sampling, we compute $\delta_{i,j}$ at the same time as we compute the sample points. We do this by storing all of the geodesic distances between each sample point p_i and all of the other vertices of $S^{(r)}$. We also compute confidence values $\omega_{i,j}$ as approximate fraction of the geodesic path between p_i and p_j that does not trace a hole of $S^{(r)}$. We use 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 from p_i to p_j on $S^{(r)}$ and where $m^h_{i,j}$ is the number of edges tracing a hole of $S^{(r)}$ on the geodesic path on $S^{(r)}$ from p_i to p_j. Since $S^{(r)}$ is a manifold, we can find the edges of $S^{(r)}$ tracing a hole of $S^{(r)}$ as edges with less than two incident faces, since every edge not adjacent to a hole of $S^{(r)}$ has two incident faces. We choose this measure for $\omega_{i,j}$, since it can be computed more efficiently than the fraction of the length of the path that does not trace the boundary of a hole. When working with data obtained from laser range scanners, $\omega_{i,j}$ is a good approximation of the fraction of the path that does not trace the boundary of a hole, because all of the edges of S have similar lengths. This choice of confidence values has been demonstrated to yield MDS embeddings that represent well the geodesic distances on $S^{(r)}$ [29].

We embed the sample vertices $P^{(r)} = \{p^{(r)}_1, \ldots, p^{(r)}_{n'}\}$ into $\mathbb{R}^k$ for a constant embedding dimension k. The embedding $X^{(r)} = \left[X^{(r)}_1, \ldots, X^{(r)}_{n'}\right]$ is found using MDS. MDS is a commonly used technique to reduce the dimensionality of high-dimensional data. Given a set of n objects $O_1, \ldots, O_n$ in d dimensions as well as the pairwise dissimilarities $\delta_{i,j}, 1 \leq i, j \leq n$ with $\delta_{i,j} = \delta_{j,i}$ between objects O_i and

O_j, the aim is to find points $X_1, \ldots, X_n$ in k dimensions with $k < d$, such that the Euclidean distance $d_{i,j}(X)$ between X_i and X_j equals $\delta_{i,j}$ for $1 \leq i, j \leq n$. This aim can be shown to be too ambitious, since in general it is not possible to find positions $X_1, \ldots, X_n$ in k dimensions such that $d_{i,j}(X) = \delta_{i,j}$ for all i, j. To find a good approximation, least-squares MDS aims to find points X_i in k dimensions, such that $E_{LS} = \sum_{i=1}^{n} \sum_{j=i+1}^{n} \omega_{i,j} \left(\delta_{i,j} - d_{i,j}(X)\right)^2$ is minimized, where $\omega_{i,j}$ are non-negative weighting coefficients with $\omega_{i,j} = \omega_{j,i}$. Since the objective function E_{LS} is a complicated function, it is easier to iteratively approximate the objective function by a simple function τ. This approach is pursued in the algorithm *Scaling by Maximizing a Convex Function* (SMACOF) that is explained by Borg and Groenen [5] and used by Elad and Kimmel [12] to compute canonical forms. We minimize τ using the *limited-memory Broyden-Fletcher-Goldfarb-Shanno* scheme [22], a quasi-Newton method, as discussed by Bronstein et al. [8] and implemented by us [29]. We perform least-squares MDS with the geodesic distances $\delta_{i,j}$ as dissimilarities and the confidence values $\omega_{i,j}$ as weights to embed the samples $P^{(r)}$ of the manifold $S^{(r)}$ to an embedding configuration $X^{(r)}$ in $\mathbb{R}^k$. The configurations $X^{(r)}$ have the property that Euclidean distances in embedding space approximate well the geodesic distances on $S^{(r)}$ according to E_{LS}. Hence, $X^{(r)}$ is called the *bending invariant form* or *canonical form* [12].

Since the embeddings $X^{(1)}$ and $X^{(2)}$ are approximately bending invariant, we can find correspondences between the sample sets by rigid registration of the canonical forms. As the canonical forms $X^{(r)}$ are invariant with respect to rotation, translation, and reflection [10], we need to consider multiple alignments of $X^{(1)}$ and $X^{(2)}$. We do this by aligning both $X^{(1)}$ and $X^{(2)}$ by their respective eigenvectors and by computing an optimal rigid correspondence using the Hungarian method for each sign assignment of the eigenvectors. After computing the 2^k different rigid correspondences, we choose the one that yields the lowest cost. A similar approach has been implemented by Jain et al. [18]. Without loss of generality assume that $n'^{(1)} \leq n'^{(2)}$. To compute the optimal rigid correspondence for a given sign assignment, we aim to find an assignment function $a(i)$ that assigns exactly one point $X^{(2)}_{a(i)}$ to every point $X^{(1)}_i$, such that $E_H = \sum_{i=1}^{n'^{(1)}} d(X^{(1)}_i, X^{(2)}_{a(i)})$ is minimized, where d denotes the Euclidean distance in $\mathbb{R}^k$. This problem can be viewed as an assignment problem and it can be solved in cubic time using the Hungarian algorithm [24]. After assigning a point $X^{(2)}_{a(i)}$) to a point $X^{(1)}_i$, we can measure $e = d(X^{(1)}_i, X^{(2)}_{a(i)})$. As e is measured as Euclidean distance in embedding space, e is an approximation of the error made by corresponding $X^{(1)}_i$ to $X^{(2)}_{a(i)}$ measured as geodesic distance on $S^{(1)}$ or $S^{(2)}$.

To eliminate erroneous matchings that assign a point $X^{(2)}_{a(i)}$ to a point $X^{(1)}_i$ although the part corresponding to $p^{(1)}_i$ is missing in $S^{(2)}$, we eliminate matching points where e is larger than the sampling resolution along the mesh.

Note that any rigid or almost rigid registration algorithm can be used to find the coarse correspondence. Jain et al. [18] use a modified iterative closest point method

for this step. As there is a one-to-one correspondence between $P^{(r)}$ and $X^{(r)}$, the correspondences of the points $P^{(1)}$ and $P^{(2)}$ directly follow.

4.2. *Fine Correspondence*

This section describes how to compute the fine correspondence given the coarse correspondence. We model the deformation of $X^{(1)}$ to $X^{(2)}$ using an approximate thin-plate spline. The following description is due to Dryden and Mardia [11]. Denote the matrix of points of $X^{(1)}$ that have a valid corresponding point in $X^{(2)}$ by $T = \left[\vec{t}_1\vec{t}_2\dots\vec{t}_l\right]$ with $l \leq n'^{(1)}$. Furthermore, denote the matrix of points of $X^{(2)}$ corresponding to T in that order by $Y = [\vec{y}_1\vec{y}_2\dots\vec{y}_l]$. The thin-plate spline deformation is $\Phi(\vec{t}) = \vec{c} + A\vec{t} + W^T s(\vec{t})$, where $\vec{c}$ is a k-dimensional vector, A is a $k \times k$ matrix, W is a $l \times k$ matrix, and $s(\vec{t}) = \left[\phi(\vec{t} - \vec{t}_1)\dots\phi(\vec{t} - \vec{t}_l)\right]^T$ is a l-dimensional vector with

$$\phi(\vec{t}) = \begin{cases} \left\|\vec{t}\right\|^2 \log\left\|\vec{t}\right\|, \left\|\vec{t}\right\| > 0, \\ 0, otherwise \end{cases}.$$

We find $\Phi(\vec{t})$ by solving the linear system of equations

$$\begin{bmatrix} S + \lambda I & \mathbf{1} & T^T \\ \mathbf{1}^T & 0 & 0 \\ T & 0 & 0 \end{bmatrix} \begin{bmatrix} W \\ c^T \\ A^T \end{bmatrix} = \begin{bmatrix} Y^T \\ 0 \\ 0 \end{bmatrix},$$

where I is the identity matrix, $\mathbf{1}$ is an $l \times 1$ vector containing 1 at each position, and λ is a constant. Following Jain et al. [18], we choose λ as the median distance between matching point pairs. The approximate thin-plate spline gives us a way of transforming any point from the embedding $X^{(1)}$ to the embedding $X^{(2)}$.

We establish the fine correspondence, that is, the correspondences between vertices in $S^{(1)} \setminus P^{(1)}$ and $S^{(2)} \setminus P^{(2)}$. To compute the correspondence in embedding space, we add all of the points in $S^{(r)} \setminus P^{(r)}$ to the embedding $X^{(r)}$ for $r = 1, 2$ [29]. To add $p^{(r)}_{n'^{(r)}+j}, j > 0$ to the least-squares MDS embedding, we compute the geodesic distances $\delta_{n'^{(r)}+j,1}, \dots, \delta_{n'^{(r)}+j,n'^{(r)}}$ as dissimilarities and the confidence values $\omega_{n'^{(r)}+j,1}, \dots, \omega_{n'^{(r)}+j,n'^{(r)}}$ as weights as before. We then aim to find a position $X^{(r)}_{n'^{(r)}+j}$ such that

$$E^*_{LS} = \sum_{i=1}^{n'^{(r)}} \omega_{n'^{(r)}+j,i} \left(\delta_{n'^{(r)}+j,i} - d_{n'^{(r)}+j,i}(X)\right)^2$$

is minimized. The function E^*_{LS} can be minimized using a limited-memory Broyden-Fletcher-Goldfarb-Shanno quasi-Newton approach [22], [29]. Note that the geodesics and weights do not need to be recomputed, since we computed and stored those values during farthest point sampling. Let the embedded points be denoted by $X^{(r)}_{n'^{(r)}+1}, \dots, X^{(r)}_{n^{(r)}}$. Once the embedded points are known, we use the approximate thin-plate spline to find the correspondences in bending invariant space. We first

compute $\tilde{X}_i^{(1)} = \Phi(X_i^{(1)})$ for $i = n'^{(1)} + 1, \ldots, n^{(1)}$. We choose $X_j^{(2)}, j = n'^{(2)} + 1, \ldots, n^{(2)}$ as corresponding point of $X_i^{(1)}$ if $\tilde{X}_i^{(1)}$ and $X_j^{(2)}$ are nearest neighbors.

As in the case of coarse correspondences, we eliminate matching points where the Euclidean distance between the points is larger than the sampling resolution along the mesh to reduce the number of erroneous matchings.

Note that this heuristic algorithm is similar, albeit more general than the work by Jain et al. [18]. Jain et al. use a slight modification of step 2 described in Algorithm 1 to register two surfaces $S^{(1)}$ and $S^{(2)}$. By deriving a coarse-to-fine approach for the correspondence problem using canonical forms, we make the use of canonical forms applicable to real-life data sets with tens of thousands of vertices. The reason is that the coarse-to-fine approach overcomes the quadratic time and storage complexity of the algorithm. A detailed analysis of the presented algorithm follows.

5. Analysis of the Algorithm

We analyze the total space and time complexity of the algorithm by analyzing each step given in Algorithm 1. Let $n = \max(n^{(1)}, n^{(2)}), n' = max(n'^{(1)}, n'(2))$. The Voronoi sampling in step 1 of the algorithm takes $O(n'n \log n)$ time and space. We can store the dissimilarities and weights computed during the execution of the sampling algorithm. That way, all the weights and dissimilarities needed during the execution of the algorithm are precomputed.

The coarse correspondence in step 2 of the algorithm takes $O(tn'^2)$ time to compute the canonical forms of $P^{(r)}$, where t is the maximum number of iterations required by the quasi-Newton algorithm. Trying all align assignments of the eigenmodes and executing the Hungarian method each time takes $O(2^k n'^3)$ time. Hence, the total time in step 2 is $O(n'^2(t + 2^k n'))$.

Computing the RBF in step 3 of the algorithm requires solving a linear system of equations of size $n' + k + 1$. This takes $O((n' + k)^3)$ time [25].

Finally, step 4 of the algorithm takes $O(tkn)$ time to project all the points in $S^{(r)} \setminus P^{(r)}$ to the embedding $X^{(r)}$, where t is the maximum number of iterations required by the quasi-Newton algorithm. This holds since all the weights and dissimilarities are precomputed. Evaluating the RBF for all of the points takes $O(nk)$ time. It remains to find the nearest neighbor in $X^{(2)}$ for each point in $\tilde{X}^{(1)}$. We use a kd-tree on $X^{(2)}$ to perform this step more efficiently. The tree can be built in $O(n \log n)$ time and uses $O(n)$ storage. Using the kd-tree, it takes $O(n\sqrt{n})$ to find all the nearest neighbors. The total time consumed by step 4 is therefore $O(n(tk + \sqrt{n}))$.

Hence, the total time complexity of the algorithm is $O(n'n \log n + n'^2(t + 2^k n') + (n' + k)^3 + n(tk + \sqrt{n}))$.

Since k is a constant and since in the average case $n' << n$, the running time becomes $O(n\sqrt{n})$. The total space complexity is $O(n'^2 + n)$, which is $O(n)$ for the average case $n' << n$. Hence, we reduced the average asymptotic space and time requirement compared to the algorithm suggested by Jain et al. [18], thereby making

the technique scalable.

Note that the running time of our algorithm for an average case is dominated by finding all the nearest neighbors. We can improve the asymptotic running time of this step, thereby improving the average asymptotic running time of our algorithm. The exact neighbors in $\mathbb{R}^k$ can be found in $O(k^{O(1)} \log n)$ time [23], [9]. This reduces the running time of our algorithm on average to $O(nk^{O(1)} \log n)$. However, the space requirement of the data structures used to answer the nearest neighbor queries grows exponentially in k.

If we allow approximate nearest neighbors, the time and space complexity of our algorithm can be improved. Let the nearest neighbor of point $\tilde{X}_i^{(1)}$ be denoted by $X_j^{(2)}$. An approximate nearest neighbor of $\tilde{X}_i^{(1)}$ with approximation constant $1+\epsilon, \epsilon > 0$ is defined as a point in $X^{(2)}$ with distance at most $(1+\epsilon)d(\tilde{X}_i^{(1)}, X_j^{(2)})$ from $\tilde{X}_i^{(1)}$. An approximate nearest neighbor of $\tilde{X}_i^{(1)}$ can be found in $O((\frac{k \log n}{\epsilon})^{O(1)})$ time using $O(n^{1/\epsilon^{O(1)}})$ space [16], [15]. Using this technique reduces the average running time of our algorithm to $O(n(\frac{k \log n}{\epsilon})^{O(1)})$.

In our implementation, we use a kd-tree to perform the nearest neighbor search due to the simplicity of its implementation.

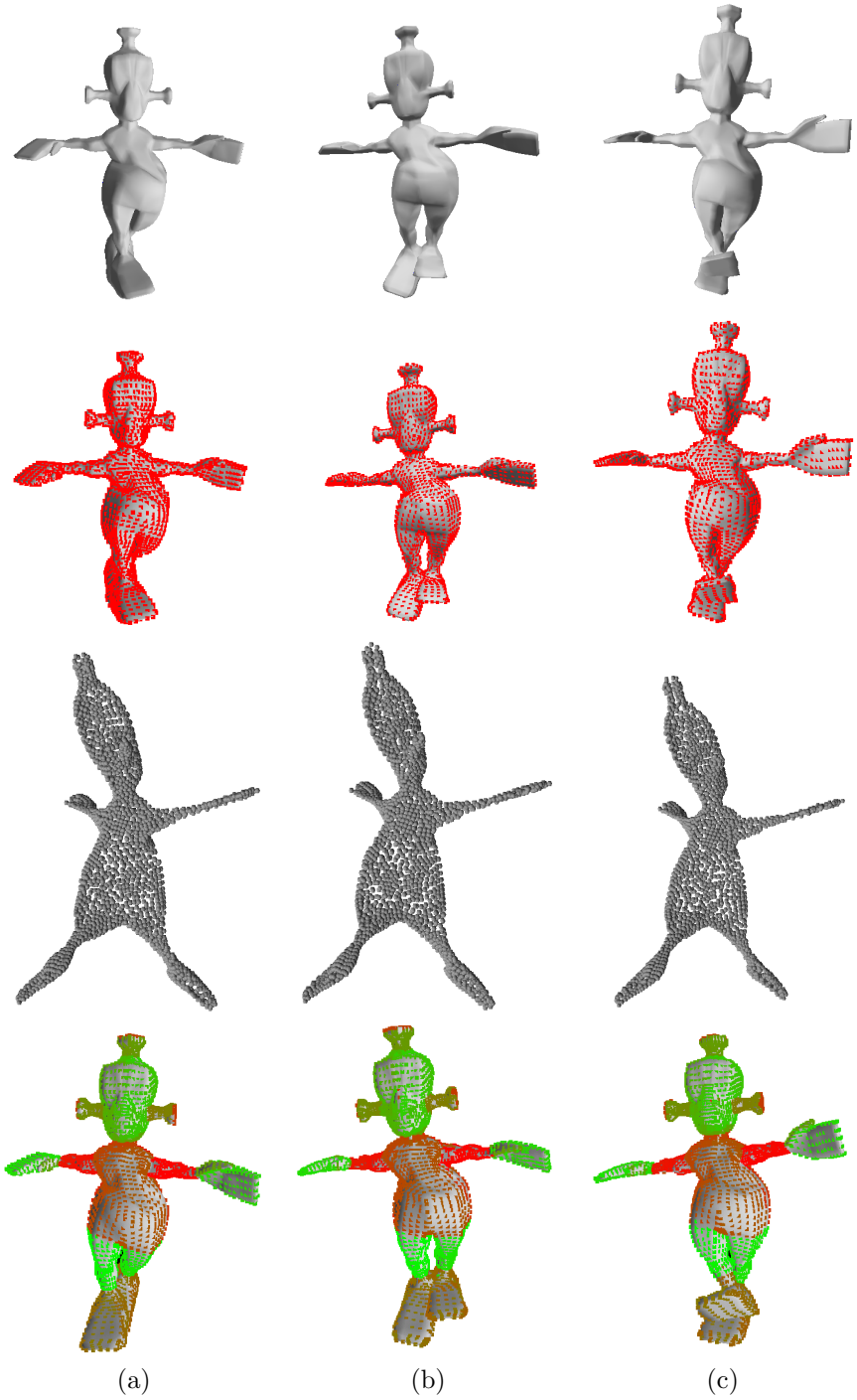


Fig. 1. *Models of alien in postures (a) to (c) with known ground truth. The first row shows the models, the second row shows the sample points used to find the coarse correspondence, the third row shows the canonical forms of the sample points, and the fourth row shows the color-coded correspondence.*

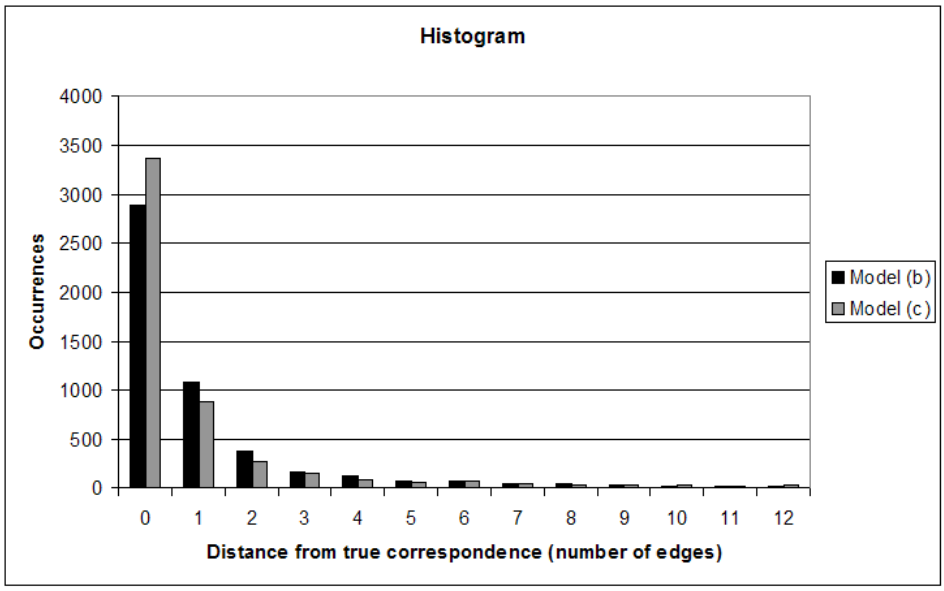


Fig. 2. *Histogram of errors made by correspondence algorithm. The black columns show the histogram of errors when corresponding posture (a) to posture (b). The grey columns show the histogram of errors when corresponding posture (a) to posture (c).*

6. Experimental Results

We implemented and tested the algorithm using models from the CAESAR database [26], the McGill 3D shape benchmark[a], and the Princeton Shape Benchmark[b]. We further used models with known ground truth to evaluate the quality of the approach.

We first describe the experiment that evaluates the quality of our approach. We chose a model of an alien from the Princeton Shape Benchmark, subdivided the model to obtain a high resolution mesh, and animated the model to obtain multiple postures with known correspondences using the automatic technique by Baran and Popović [4]. Figure 1 illustrates the experiment. The three postures of the alien used to conduct the experiment consist of 6858 vertices and are shown in the first row of the figure. The second row of the figure shows the 2500 sample vertices found on each model using farthest point sampling. The third row of the figure shows the canonical forms of the sample points. The fourth row shows the full fine correspondence found by the algorithm. We found the correspondences between posture (a) and postures (b) and (c) in this experiment. Each vertex is assigned

[a]http://www.cim.mcgill.ca/ shape/benchMark/
[b]http://shape.cs.princeton.edu/benchmark/

a unique color between red and green in posture (a). The corresponding points in postures (b) and (c) are then displayed using the same color in row four of Figure 1. We can see that a visually pleasing correspondence is found.

We compare the correspondences found by our algorithm to the ground truth by computing the geodesic distances between the correspondence found by the algorithm and the true correspondence for each vertex. We measure the error in correspondence as the number of edges along the shortest path between the correspondence found by the algorithm and the true correspondence. Since the algorithm rejects erroneous matchings automatically as outlines in Section 4.1, some points do not obtain a correspondence. We do not assign an error to rejected correspondences. When registering posture (a) and posture (b), 1935 correspondences are rejected as erroneous. When registering posture (a) and posture (c), 1842 correspondences are rejected as erroneous. A histogram of the error encountered is shown in Figure 2. The histogram shows two different data sets: the set of errors when corresponding posture (a) to posture (b) is shown in black and the set of errors when corresponding posture (a) to posture (c) is shown in grey. Nearly all of the correspondences found by our algorithm are accurate within a distance of two edge lengths. This shows that the approach taken in this paper yields correspondences of high quality.

For some poses of the alien model, the correspondences found are erroneous due to symmetry alignment. A case for which this problem occurs is when registering the posture shown in Figure 1 (a) with the posture shown in Figure 3. In this case, the left side of the posture shown in Figure 1 (a) is found to correspond to the right side of the posture shown in Figure 3 and vice versa. To illustrate this, Figure 3 shows a true correspondence we aim to find in green and the correspondence found by the algorithm in red. The problem of symmetric alignments is a limitation of our approach and it is discussed in more detail below.

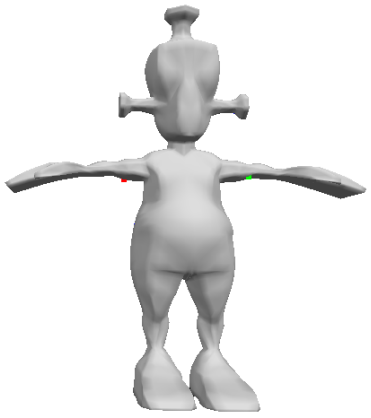
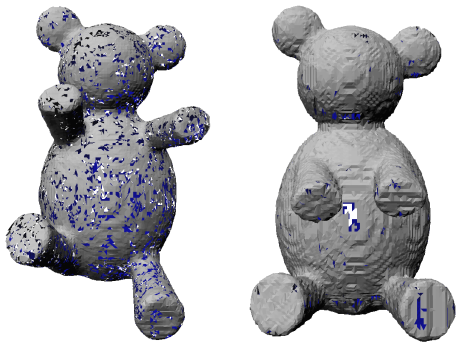


Fig. 3. *Erroneous correspondence due to symmetry problems.*

We further conducted two experiments with incomplete real-life data sets. The first articulated model we experimented with is the model of a teddy bear from the McGill 3D shape benchmark. Figure 4(a) shows the first model consisting of 21338 vertices, 2000 of which were chosen as sample points. Figure 4(b) shows the second model consisting of 25658 vertices, 2050 of which were chosen as sample points. Note that both models contain many small holes. Figure 5 shows the aligned canonical forms of the two bears. The canonical forms are almost rigid transformations. The correspondence computed by our algorithm is visualized in Figure 6. We visualize the result by assigning a unique color to each vertex on the first model and by drawing vertices on the second model in same color as their corresponding vertices.



(a) (b)

Fig. 4. *Models of teddy bears from the McGill 3D shape benchmark.*

The second articulated model we experimented with is a subject of the CAESAR data base. The models of the CAESAR data base are available in three different postures. The three original models are shown in Figures 7(a) to (c). We manually changed the models to exclude the chair and fix large holes on the back for models (b) and (c). For model (b), the hands were detached from the legs, since our method cannot cope with changing topologies. Furthermore, we excluded one hand for each of the models to avoid symmetry problems during registration. The changed models are shown in Figures 8(a) to (c) and the modified models contain about 30,000 vertices. The sample sizes were chosen between 2,000 and 3,000 vertices.

Figure 9 shows the results of registering the models using the proposed algorithm. Figure 9(a) shows the registration of model (a) with model (c). We visualize the result by segmenting model (a), by assigning each segment a unique color, and by drawing vertices on the model (c) in same color as their corresponding vertices. We can see that the overall correspondence is correct, although some vertices of the head in (a) correspond to the left arm in (c). Figure 9(b) shows the registration of model (b) with model (c). We visualize the result by using the segmentation of model (c), by assigning each segment a unique color, and by drawing vertices on the

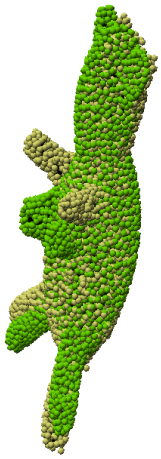


Fig. 5. *Optimal rigid alignment of canonical forms.*

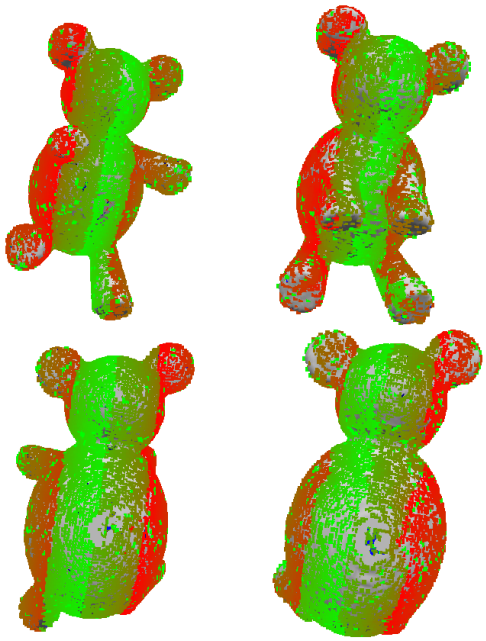
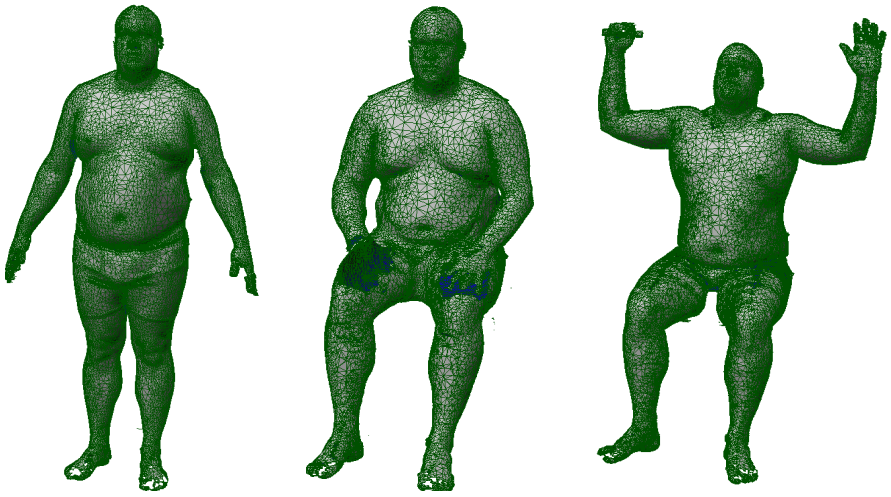


Fig. 6. *Fine correspondence computed by the algorithm.*

model (b) in same color as their corresponding vertices. It is clearly visible that the correspondence registered the left side of model (c) with the right side of model (b). This happens although we excluded one hand from consideration to avoid problems



(a) (b) (c)

Fig. 7. *Models of CAESAR data base.*

(a) (b) (c)

Fig. 8. *Modified models of CAESAR data base.*

of this kind.

All the experiments shown in this Section were conducted on a Pentium(R) D with 3.0 GHz and 3.5 GB of RAM. Since the emphasis of this paper is on the fact that our algorithm is a theoretically more efficient approach than the approaches presented by Bronstein et al. [7] and Jain et al. [18], we developed a non-optimized

(a) (b)

Fig. 9. *Fine correspondence computed by the algorithm.*

experimental implementation. Hence, the running time of the experiments were just under twenty minutes for the largest of the data sets. Those running times could be improved by using multi-threading and by implementing some of the algorithms on the GPU as in Bronstein et al. [7].

7. Conclusion

This section summarizes the contribution of this work by comparing our algorithm to the algorithms proposed by Jain et al. [18] and to recent related work by Bronstein et al. [7]. Furthermore, we summarize limitations of our algorithm.

Our algorithm works well for non-symmetric surfaces which can be represented well in Euclidean spaces. The main advantage of our algorithm is that no prior knowledge about the objects being registered is required. The approach presented in this paper is more time and space efficient than the approach by Jain et al. due to our coarse-to-fine strategy. While Jain et al.'s approach was a conceptual contribution, all of the experiments conducted were on small-scale models consisting of about 250 vertices. Our approach extends this concept and makes it applicable to real-life data sets with tens of thousands of vertices by overcoming the quadratic

time and space complexity using a coarse-to-fine strategy.

The approach presented in this paper can be viewed as a variation of the approach by Bronstein et al. Both approaches were developed independently at the same time. The main difference between the two approaches is the choice of the embedding space. While we embed to $\mathbb{R}^k$, Bronstein et al. use generalized multidimensional scaling to embed one manifold into another. While the approach of Bronstein et al. is more suitable to embed spaces that are not flat such as surfaces of human bodies, the approach assumes that one of the meshes is a complete mesh or a template mesh. Our approach does not assume this prior knowledge. Therefore, our approach is favorable for manifolds with nearly flat intrinsic geometry, where both models being registered contain significant amounts of missing data. Another limitation of Bronstein et al.'s approach is its high time complexity. To compute the canonical form, our approach recomputes the Euclidean distances $d_{i,j}(X)$ along with the gradient of $d_{i,j}(X)$ with respect to X in each SMACOF iteration. The approach by Bronstein et al. needs to compute geodesic distances on the template mesh to replace $d_{i,j}(X)$. This means that in each SMACOF iteration, geodesic distances along with the gradient of the geodesic distances with respect to the positions of the points on the template surface need to be computed. Hence, while our algorithm takes $O(tn'^2)$ time to compute the canonical form of n' sample points in t iterations, the approach by Bronstein et al. takes $O(tn'^2 \log n')$ time. This makes our approach more efficient and therefore more suitable for large data sets. While our approach finds dense point-to-point correspondences for data sets with tens of thousands of vertices, the approach by Bronstein et al. was only shown to compute correspondences between coarse sample sets containing about 3000 vertices.

Finally, we summarize some limitations of our approach that should be addressed in the future:

- Surfaces that cannot be represented well in Euclidean spaces cannot be registered reliably using this algorithm. The human shapes in the experimental section show that although the shape itself is not symmetric, the canonical forms become symmetric due to the embedding error.
- Symmetric surfaces may be registered wrong by the coarse correspondence step of the algorithm.
- Surfaces with large holes cannot be registered reliably using this algorithm, since large holes alter the global shape of the canonical embedding of the sample points. This is the reason we fixed the back parts of the sitting human models.
- Surfaces with many significant outliers cannot be registered reliably using this algorithm, because MDS is not robust with respect to outliers [10]. This means that outliers can alter the global shape of the canonical embedding of the sample points.

References

1. Brett Allen, Brian Curless, and Zoran Popović. The space of human body shapes: reconstruction and parameterization from range scans. *ACM Transactions on Graphics*, 22(3):587–594, 2003.
2. Dragomir Anguelov, Praveen Srinivasan, Daphne Koller, Sebastian Thrun, Hoi-Cheung Pang, and James Davis. The correlated correspondence algorithm for unsupervised registration of nonrigid surfaces. In *Neural Information Processing Systems*, 2004.
3. Dragomir Anguelov, Praveen Srinivasan, Daphne Koller, Sebastian Thrun, Jim Rodgers, and James Davis. Scape: shape completion and animation of people. *ACM Transactions on Graphics*, 24(3):408–416, 2005.
4. Ilya Baran and Jovan Popović. Automatic rigging and animation of 3d characters. *ACM Transactions on Graphics*, 26(3), 2007.
5. Ingwer Borg and Patrick Groenen. *Modern Multidimensional Scaling Theory and Applications*. Springer, 1997.
6. Alexander M. Bronstein, Michael M. Bronstein, and Ron Kimmel. Generalized multidimensional scaling: a framework for isometry-invariant partial surface matching. *National Academy of Sciences*, 103(5):1168–1172, 2006.
7. Alexander M. Bronstein, Michael M. Bronstein, and Ron Kimmel. Calculus of non-rigid surfaces for geometry and texture manipulation. *IEEE Transactions of Visualization and Computer Graphics*, 13(5):902–913, 2007.
8. Alexander M. Bronstein, Michael M. Bronstein, Ron Kimmel, and Irad Yavneh. Multigrid multidimensional scaling. *Numerical Linear Algebra with Applications, Special issue on multigrid methods*, 13(2–3):149–171, 2006.
9. Kenneth L. Clarkson. A randomized algorithm for closest-point queries. *SIAM Journal on Computing*, 17(4):830–847, 1988.
10. Trevor Cox and Michael Cox. *Multidimensional Scaling, Second Edition*. Chapman & Hall CRC, 2001.
11. Ian Dryden and Kanti Mardia. *Statistical Shape Analysis*. Wiley, 2002.
12. Asi Elad and Ron Kimmel. On bending invariant signatures for surfaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(10):1285–1295, 2003.
13. Yuval Eldar, Michael Lindenbaum, Moshe Porat, and Yehoshua Zeevi. The farthest point strategy for progressive image sampling. *IEEE Transactions on Image Processing*, 6(9):1305–1315, 1997.
14. John C. Gower. Some distance properties of latent root and vector methods used in multivariate analysis. *Biometrika*, 53:325–338, 1966.
15. Piotr Indyk. Dimensionality reduction techniques for proximity problems. In *Symposium on Discrete Algorithms*, pages 371–378, 2000.
16. Piotr Indyk. *Nearest Neighbors in High-Dimensional Spaces.* In: Jacob E. Goodman and Joseph O'Rourke (Editors). Handbook of Discrete and Computational Geometry (2nd edition). Chapter 39. CRC Press, 2004.
17. Varun Jain and Hao Zhang. A spectral approach to shape-based retrieval of articulated 3d models. *Computer-Aided Design*, 39(5):398–407, 2007.
18. Varun Jain, Hao Zhang, and Oliver van Kaick. Non-rigid spectral correspondence of triangle meshes. *International Journal on Shape Modeling*, Special Issue of SMI 2006, to appear.
19. Ron Kimmel and James Sethian. Computing geodesic paths on manifolds. *National Academy of Sciences*, 95:8431–8435, 1998.
20. Vladislav Kraevoy, Alla Sheffer, and Craig Gotsman. Matchmaker: Constructing constrained texture maps. *ACM Transactions on Graphics*, 22(3):326–333, 2003.
21. Xinju Li and Igor Guskov. Multi-scale features for approximate alignment of point-

based surfaces. In *Proceedings of 3rd ACM Siggraph/Eurographics Symposium on Geometry Processing*, 2005.
22. Dong C. Liu and Jorge Nocedal. On the limited memory method for large scale optimization. *Mathematical Programming*, 45:503–528, 1989.
23. Stefan Meiser. Point location in arrangements of hyperplanes. *Information and Computation*, 106(2):286–303, 1993.
24. James Munkres. Algorithms for the assignment and transportation problems. *Journal of the Society of Industrial and Applied Mathematics*, 5(1):32–38, 1957.
25. Willian Press, Brian Flannery, Saul Teukolsky, and William Vetterling. *Numerical Recipes in C: the Art of Scientific Computing*. Cambridge University Press, 1993.
26. Kathleen Robinette, Hans Daanen, and Eric Paquet. The caesar project: A 3-d surface anthropometry survey. In *3-D Digital Imaging and Modeling*, pages 180–186, 1999.
27. John Schreiner, Arul Asirvatham, Emil Praun, and Hugues Hoppe. Inter-surface mapping. In *ACM Transactions on Graphics*, volume 23, pages 870–877, 2004.
28. Sébastien Valette and Jean-Marc Chassery. Approximated centroidal voronoi diagrams for uniform polygonal mesh coarsening. *Computer Graphics Forum (Eurographics 2004 proceedings)*, 23(3):381–390, 2004.
29. Zouhour Ben Azouz, Prosenjit Bose, Chang Shu, and Stefanie Wuhrer. Approximations of geodesic distances for incomplete triangular manifolds. In *Proceedings of the 19th Canadian Conference on Computational Geometry*, pages 177–180, 2007.