Robust 3D Shape Correspondence in the Spectral Domain

Varun Jain Hao Zhang
Graphics, Usability and Visualization (GrUVi) Lab
School of Computing Science
Simon Fraser University, Burnaby, BC, Canada
vjain,haoz@cs.sfu.ca

Abstract

We present an algorithm for finding a meaningful vertex-to-vertex correspondence between two 3D shapes given as triangle meshes. Our algorithm operates on embeddings of the two shapes in the spectral domain so as to normalize them with respect to uniform scaling and rigid-body transformation. Invariance to shape bending is achieved by relying on geodesic point proximities on a mesh to capture its shape. To deal with stretching, we propose to use non-rigid alignment via thin-plate splines in the spectral domain. This is combined with a refinement step based on the geodesic proximities to improve dense correspondence. We show empirically that our algorithm outperforms previous spectral methods, as well as schemes that compute correspondence in the spatial domain via non-rigid iterative closest points or the use of local shape descriptors, e.g., 3D shape context.

1. Introduction

Given two 3D shapes represented as 2-manifold triangle meshes, the correspondence problem seeks to establish a meaningful mapping between them. The mapping can be between the two sets of mesh vertices, between two coarse sets of feature points selected on the meshes, or a continuous one between all points on the two manifolds. This is a fundamental problem in computer graphics and shape modeling, with such applications as texture mapping [20], mesh morphing [1], shape registration, and object recognition.

A recent trend in research into the mesh parameterization problem, which essentially computes a dense surface correspondence, is to look for effective techniques to construct a *cross parameterization* between two meshes directly [21, 29], i.e., without relying on an simple common parameter domain. Such a parameterization allows one to obtain compatible connectivity among a set of models in a feature-sensitive way [21, 27], greatly facilitating tasks such as shape blending and various forms of attribute transfer. A crucial first step for constructing a cross parameterization is the identification of a sparse set of matching feature points on the two shapes. Most methods [27, 20, 21, 29] rely on the user to specify the initial feature correspondence manually. It is well known that even if the sets of feature points have been given, finding a meaningful correspondence automatically with robustness to both rigid and non-rigid geometric transformations is notoriously difficult. This is the problem we wish to address in this paper.

Given two 2-manifold triangle meshes, possibly with different vertex counts, we compute a correspondence between the mesh vertices. The spatial coordinates of the mesh vertices are first converted into two affinity matrices, which are obtained by applying a *Gaussian kernel* to the pairwise *geodesic distance* matrices for the two meshes. This way, each vertex of a mesh is represented using intrinsic structural information. The correspondence is obtained by matching points based on this information and this is carried out in a k-dimensional *spectral* domain via eigenspace projections. Typically, k is much smaller than the size of the affinity matrices. Thus the dimensionality of the structural information has been effectively reduced.

Appropriate choice of the Gaussian kernel width and the use of spectral embeddings described above ensure that our matching procedure is invariant to rigid body transformations, uniform scaling, and shape bending. This is because the affinities we use are invariant to precisely these transformations. We also propose to scale the eigenvectors by the square root of the eigenvalues in forming the spectral embeddings so as to achieve robustness against difference in mesh vertex counts and choice of the dimensionality of the embeddings. However, our experiments show that matching based solely on such embeddings, e.g., using the L_2 metric [31], can be non-robust to stretching in the shapes. To this end, we perform a non-rigid alignment in the spectral domain using thin-plate splines. A refinement step using the original geodesic proximity data enhances the performances of the algorithm in the case of

dense correspondence. Through formal arguments and numerous experiments, we demonstrate that our approach outperforms previous methods which operate on spectral embeddings [7, 8, 31, 33], as well as schemes that find correspondence in the spatial domain via non-rigid iterative closest points [6] or the use of local shape descriptors, such as shape contexts [3] and curvature maps [13].

The rest of the paper is organized as follows. Section 2 gives a brief survey of previous work. Section 3 gives an overview of the problem in question and our algorithm; it also establishes some notations used in subsequent sections. In Section 4, we describe the construction of the spectral embeddings in more details, argue for its various properties, and demonstrate its non-robustness to stretching. We also explain our eigenvalue scaling scheme and its effect on the spectral embeddings. In Section 5, we present the major components of our correspondence algorithm. This is followed by experimental results in Section 6, where we also point out some limitations of our current approach. Finally, we conclude in Section 7 and comment on future work.

2. Previous work

Point correspondence may be computed based on either absolute coordinates or relative information, e.g., using weighted graphs. Two main classes of methods exist for spatial correspondence: those using local shape descriptors and those relying on iterative alignment schemes. The first class of techniques describe every point on a shape by encoding shape information from the perspective of that point. Point matching is then based on appropriate distances between the descriptors. Well-known descriptors for images include shape contexts [3] and spin images [17], both utilizing a histogram obtained by binning the space around a point according to the Euclidean metric and collecting point counts. These methods have subsequently been generalized in a straightforward manner to handle 3D point sets [18].

Neither shape contexts nor spin images are invariant to shape bending. A promising remedy is the curvature maps of Gatzke et al. [13]. Local shape descriptors are constructed on a mesh by dividing the geodesic neighborhood of a vertex into bins. Although geodesic binning is invariant to bending (not stretching), the histograms computed are based on curvature distributions, which, even if estimated robustly, are not invariant to bending.

More recent shape signature based methods include those by Gelfand et al. [14] and Li et al. [23]. Both methods are robust under only rigid transformations. The partial matching scheme of the former do provide a way to detect articulated subparts of a shape and subsequently match them to the subparts of the second shape. However, their work lacks a discussion and analysis of the performance and robustness of the matching method when there are multiple

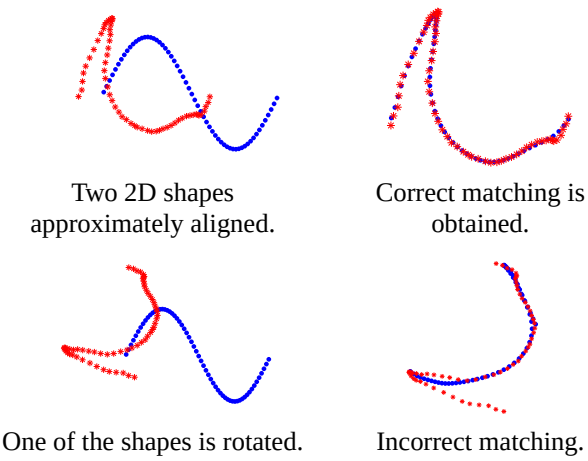


Figure 1. TPS-RPM [6] failed by rotation.

pose changes in the models or when the models consist of a large number of articulated subparts.

Iterative alignment schemes compute a correspondence and a transformation which would transform one shape into another at each step. The correspondence is usually based on a "closest point" criteria and the transformation is obtained by optimizing an energy. Such techniques include the well known iterative closest point (ICP) algorithm of Besl and Mckay [4] and its variants [28], which can handle affine transformations. Recent works, most notably the TPS-RPM method of Chui and Rangarajan [6], attempt to incorporate non-rigid deformations into the ICP framework, using thin-plate splines to model the deformation. However, these methods can easily get trapped in bad local minima if the shapes are not approximately aligned initially, since the correspondence, which dictates the optimization, is computed using an Euclidean closest point method. Thus rotation alone can cause a bad matching, as shown in Figure 1.

Sumner et al. [32] and Zayer et al. [34] attempt to alleviate this problem by fixing a small number of feature points on the shapes to be matched. Sumner et al. rely on these feature points as guidance to ICP in order to escape local minima, whereas Zayer et al. use interpolation, based on barycentric coordinates, of the correspondence between the feature points to compute the remaining point correspondences. In both methods, it is imperative that the feature points selected on the two meshes be corresponding in a meaningful way. This can only be done with user assistance as automatic selection and matching of the feature points is equivalent to the correspondence problem we are trying to solve in the first place.

Spectral shape correspondence involves first constructing intrinsic (relative) point representations of the two shapes, in the form of weighted graph adjacency or affinity matrices. Elad and Kimmel [10] make use of geodesic

proximities to construct bending-invariant surface signatures through multi-dimensional scaling. Application to object classification has been considered, but they do not solve the harder correspondence problem. Given a proximity matrix, a k-dimensional spectral embedding can be computed via principal component analysis (PCA). Shapiro and Brady [31] use L_2 distances between the embedded points to compute a correspondence, while Umeyema [33] chooses the correlation between the embedding coordinates. Both Caelli and Kosinov [5] and Carcassoni and Hancock [8] rely on spectral clustering and cluster correspondence to guide point correspondence. The use of spectral embeddings has traditionally been exploited in the computer vision and machine learning literature. Recently, they have found several applications in geometry processing as well, including mesh segmentation [22], spherical parameterization [16], and surface reconstruction [19].

Common to all the existing spectral correspondence techniques is the premise that the eigenmodes from two similar shapes should match up, according to the magnitude of their corresponding eigenvalues. One of our main observations is that this ordering of the eigenmodes is not always reliable. As the eigenvalues characterize data variance in the direction of the corresponding eigenvectors, eigenmode ordering based on eigenvalues implies ordering by data variance. This may not be appropriate since variance only captures global information and does not reflect the way specific data points would vary. We observe that under shape stretching, certain eigenmodes may be "switched". Failure to resolve such reflections or other non-rigid discrepancies between spectral embeddings will lead to poor matching results. In this paper, we propose heuristics to handle such switchings in the spectral domain, where the spectral embeddings will be corresponded after alignment using non-rigid ICP based on thin-plate splines.

3. Overview

Let us first give a brief overview of the problem we address and the algorithm we propose. Given two 3D shapes M_1 and M_2, in the form of triangular meshes and with n_{M_1} and n_{M_2} vertices, respectively, we wish to compute a correspondence C between the two sets of vertices in M_1 and M_2. That is, $C(i)$ is the vertex in M_2 that best corresponds to vertex i in M_1. Note that the correspondence computed is not required to be bijective. However, in the case where $n_{M_1} = n_{M_2}$ and a one-to-one correspondence is sought, we can easily modify our method to meet the goal.

Our method of computing C is as follows: first, we establish an $n_{M_1} \times n_{M_1}$ affinity matrix A where A_{ij} is the affinity between vertices i and j of M_1. Similarly, we compute an $n_{M_2} \times n_{M_2}$ matrix B, the affinity matrix for M_2. The affinities that we use in our implementation are based on geodesic distances in order to attain invariance to bending. Next we find the spectral embeddings $\hat{A}_k$ and $\hat{B}_k$ of the matrices A and B, respectively. These embeddings give k-dimensional coordinates of all the vertices of M_1 and M_2. The embeddings are based on the eigenvectors of A and B, properly processed as we describe in Section 4. The purpose of transforming the 3D mesh from the spacial domain to the k-D spectral domain is to attain invariance to bending, rigid transformations, and uniform scaling, as well as robustness to difference in mesh sizes, for example.

Once we have the two k-D meshes that are already properly normalized, we use iterative alignment to robustly align them and then obtain a correspondence via best matching based on the L_2 distance. As we shall explain in Section 4, rigid alignment is insufficient to deal with moderate stretchings in the shapes to be matched. Hence, we modify the well-known iterative closest point (ICP) algorithm to include non-rigid transformations. Specifically, we use thin plate splines to model non-rigid transformations and compute a registration between the meshes in the spectral domain. As mentioned earlier, this registration is not required to be one-to-one, but can be forced to be bijective by using the Hungarian algorithm for bipartite matching.

4. Spectral embedding

In general, intrinsic point representations can be obtained via pairwise point proximities, specified by a symmetric *affinity matrix* $A = \{a_{ij}\}$, where $a_{ij} \geq 0$ characterizes the similarity or simply the graph adjacency [5, 33] between points i and j. One may view the affinity matrix A as a data vector whose n columns (or rows) represent n-dimensional data points.

The most common proximity measure used to define relationship between points for shape matching is the Euclidean distance [7, 8, 30, 31], which implies invariance to rotation and translation. For mesh correspondence, we use geodesic distances between the mesh vertices, computed via fast marching [10], to include invariance to bending as well. Invariance to uniform scaling is achieved by mapping the geodesic proximities into the interval $[0, 1]$ using a *scale-dependent*, *positive semi-definite* kernel function. In this paper, we use Gaussian kernels which is a common choice for spectral correspondence [7, 8, 30, 31].

Although the point proximities contain a great deal of shape information, without a proper point mapping, one cannot compare such representations for two data sets directly. Also, the size of the data sets, or the dimensionality of the point representations, may not be the same. Last but not the least, the high dimensional representations may contain a great deal of redundancy, resulting in unnecessarily high computational cost. These observations naturally lead us to consider transforming two data sets, respectively, into

Figure 2. A human mesh (left) and its 3D spectral embedding, constructed using the second, third, and fourth eigenvectors. This particular choice of the eigenvectors is explained in Section 5.

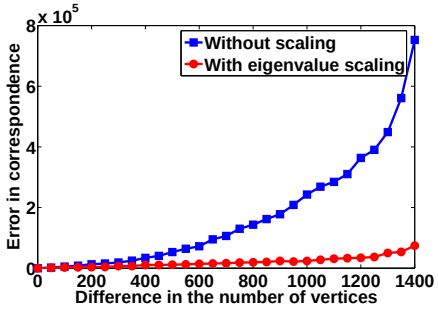


Figure 3. Effect of eigevalue scaling on correspondence, based on the correspondence error plots.

some information-preserving subspaces that share the same low dimensionality. This can be accomplished through principal component analysis (PCA) on the affinity data.

4.1. Principal component analysis

Given the data (affinity) matrix $A \in \mathbf{R}^{n\times n}$, we first compute its principal components $\mathbf{e_1}, \ldots, \mathbf{e_n}$, which are the *normalized* eigenvectors of the autocorrelation matrix $R = AA^T$. Since A is symmetric, $R = A^2$ and $\mathbf{e_1}, \ldots, \mathbf{e_n}$ are simply the eigenvectors of the affinity matrix A. Let $\lambda_1, \ldots, \lambda_n$ be the corresponding eigenvalues of A and suppose that $\lambda_1 \geq \ldots \geq \lambda_n$. Projecting the data matrix onto the first k principal components yield

$$\hat{A}_k = E_k^T A = \Lambda_k E_K^T, \qquad (1)$$

where $E_k = [\mathbf{e_1}|\ldots|\mathbf{e_k}]$, $\Lambda_k = diag(\lambda_1, \ldots, \lambda_k)$, and the columns of $\hat{A}_k$ represent a k-dimensional *spectral embedding* of the data points. Note that a point permutation induces the same permutation of the embedded coordinates but leaves the spectrum invariant.

In the case of mesh spectral embeddings, the data points are mesh vertices. A spectral embedding associates with each mesh vertex a k-dimensional coordinate. In the 3D spectral domain, one can visualize the embedding of a mesh M by rendering a mesh whose connectivity is the same as M and whose vertices are given by the embedding coordinates, as shown in Figure 2.

Although $\hat{A}_k$ gives a provably best k-dimensional approximation of A (in terms of the Frobenius norm), it may not be suitable for matching. The more important requirement is for the projection axes, derived from the principal components, to be compatible between two data sets.

4.2. Eigenvalue scaling

Given two affinity matrices A and B characterizing two shapes, possibly in different scales, a scale-dependent kernel can normalize the affinity values in A and B. If the number of vertices, n_A and n_B, in the two (mesh) shapes differ however, we first need to truncate both spectral embeddings to the same dimension $k \leq \min\{n_A, n_B\}$. In addition, since we normalize each eigenvector, the cardinality of a data set affects the magnitude of the entries in its eigenvectors, which in turn affects the embeddings.

Correspondence algorithms that use unscaled eigenmodes [7, 8, 31] as spectral embeddings are common. Shapiro and Brady [31] first suggest a scaling of the the eigenmodes by eigenvalues, as in (1), but did not elaborate. Caelli and Kosinov [5] scale the eigenmodes using *squared* eigenvalues and then project the resulting embeddings onto the unit k-sphere for matching graphs of different vertex counts. Evidently, proper scaling of the eigenmodes is a crucial normalization step. None of these approaches adequately resolves the discrepancies in the scales of the principal components (or embeddings) due to difference in the cardinality of the data sets. In this paper, we propose to scale the principal components by the *square root* of the eigenvalues, yielding projections

$$\hat{A}_k = \Lambda_k^{\frac{1}{2}} U_k^T \quad \text{and} \quad \hat{B}_k = \Gamma_k^{\frac{1}{2}} V_k^T, \qquad (2)$$

where $A = U\Lambda U^T$ and $B = V\Gamma V^T$ are the eigenvalue decompositions of A and B, respectively, with $U_k, V_k, \Lambda_k, \Gamma_k$ defined as in (1). Spectral embeddings of this form are well known in the spectral clustering literature [26].

Justifications: Consider the vector of projections $\hat{\mathbf{a}}_i$ from set A. We can estimate the scale of these projections by $s_{A,i} = ||\hat{\mathbf{a}}_i||^2/n_A$. From (2), we have

$$\hat{\mathbf{a}}_i = \sqrt{\lambda_i}\mathbf{u_i} \text{ and } \hat{\mathbf{b}}_\mathbf{i} = \sqrt{\gamma_\mathbf{i}}\mathbf{v_i}.$$

It follows that $s_{A,i} = \lambda_i/n_A$ and $s_{B,i} = \gamma_i/n_B$. With the affinity matrices having unit diagonal elements, signaling

that a point has maximal affinity to itself, we have

$$s_{A,i} = \frac{\lambda_i}{trace(A)} = \frac{\lambda_i}{\sum_{j=1}^{n_A} \lambda_j} \text{ and } s_{B,i} = \frac{\gamma_i}{\sum_{j=1}^{n_B} \gamma_j}.$$

We do not normalize these scales to some constant, since they represent data variations along the projection axes and thus contain shape information. We only wish to remove the effect of different data size; this is achieved by normalizing the eigenvalues, which represent data variations.

Another justification for (2) is that the dot-product matrices $\hat{A}_k^T \hat{A}_k$ and $\hat{B}_k^T \hat{B}_k$ are respectively the best rank-k approximations, in Frobenius norms, of A and B [9], which are already normalized to scale. Using the same argument, we see that the dot product matrices resulting from eigenmode scaling with eigenvalues themselves [31] become best rank-k approximations of the autocorrelation matrices $AA^T = A^2$ and B^2, respectively, whose entries do depend on the size of the data sets.

Experimental results: The effectiveness of our eigenvalue scaling scheme is shown in Figure 3, where we plot the correspondence errors in the case of scaled versus unscaled spectral embeddings. The correspondence error is measured as the total geodesic distance $\sum_{i=1}^{n} g(v_i, v_i')$, where n is the number of vertices to be matched, v_i is the vertex corresponding to i that is computed by an algorithm, and v_i' is the ground-truth match for i. The plots are against the difference in the number of vertices of the two meshes to be matched. The correspondence algorithm used is our own and it is described in Section 5. In producing the two plots, the only difference is whether the eigenmodes are scaled.

Measuring error for dense correspondence is not easy since the ground-truth correspondence is impractical to establish manually. In our evaluation, we first construct successively decimated copies of the same 3D mesh using the QSlim mesh decimation program of Garland [12]. Next we use our algorithm to find correspondences between the decimated copies and the original mesh and measure correspondence errors. The ground-truth can be trivially established since QSlim retains the positions of undecimated vertices.

4.3. Non-robustness of eigenmodes

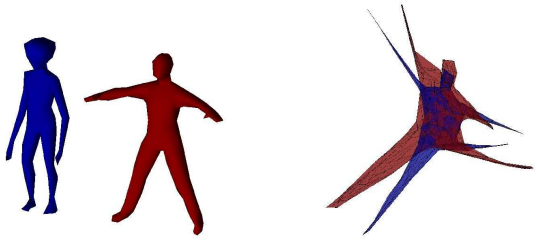
Eigenmode switching: Perturbation theory predicts that when eigenvalues move close to each other, the corresponding eigenvectors may switch order [15]. We have observed that such switching can occur early in the eigenvalue order, e.g., between 4 and 8, even when the two shapes being matched are perceptually similar. But there is no general pattern of eigenvalue clustering that is sufficiently reliable to detect the switchings. As switching of two coordinates, the eigenvectors, induces a reflection in the spectral domain, spectral correspondence based on the L_2 distance measure or correlations [31, 33], even with the aid of clustering [5, 8], can fail.

For a visual illustration of eigenmode switching, we color-plot the eigenvectors in MATLAB, where the entries in an eigenvector are used as indices into the color map. To enhance our illustration, we nonlinearly warp the color map. As shown in Figure 6, given in the color plate, two similar shapes have compatible eigenmodes, reflected by consistent color plots, only up to the 4^{th} eigenvalue. The 5^{th} and 6^{th} eigenmodes are switched and color patterns for the next a few eigenvectors, those exhibiting higher-frequency color variations, do not exhibit any discernible patterns. Evidently, correspondence analysis using eigenvalue orderings to pair up eigenmodes beyond the 4^{th} one would be hard to justify in this case.

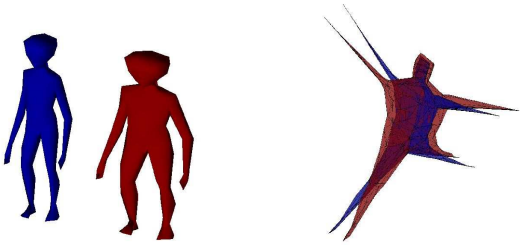
Relevance to eigenvalue scaling: One interesting point to note is that as the magnitude of the eigenvalues of the geodesic affinity matrices exhibit rapid decay, as shown in the caption of Figure 6, eigenvalue scaling has the effect of rapidly attenuating the effects of higher-frequency eigenvectors. This would be quite appropriate since these eigenvectors are less reliable to use for correspondence analysis. As a side effect, the resulting correspondence algorithm will be less sensitive to the number of eigenmodes chosen. In previous works, e.g., [8], some heuristic has to be adopted to determine the proper dimensionality to use.

Sign flips: Besides reflections induced by switching, other transformations in the spectral domain also need to be handled to achieve robust spectral correspondence. One such transformation is due to the arbitrary determination of the signs of the eigenvectors by the numerical eigenvalue program, as already noted in previous works [5, 31]. Note that this is another form of reflection. Caelli and Kosinov [5] propose to use a dominant sign correction, always ensuring that there are more positive entries in each eigenvector. This is highly unreliable however since in practice, most eigenvectors have about the same number of positive and negative entries. Shapiro and Brady [31] use a greedy approach to correct one sign at a time by optimizing for a correspondence cost. In the presence of eigenmode switchings, this approach is not robust either.

Other transformations: Consider the spectral embeddings of two similar human meshes using the 2^{nd}, 3^{rd} and 4^{th} eigenvectors (this particular choice of the eigenvectors is explained in Section 5), as shown in Figure 4(a). Ideally, the embeddings would be perfectly aligned. However, a rotational difference in the embeddings is clearly visible. In addition, there are also other discrepancies of a non-rigid nature. Another example is given in Figure 4(b), where the second (light gray) mesh is merely a scaled version (scaled along the x direction) of the first (dark gray) mesh. But there again is a rotation in the embedding. We be-



(a) Two similar human meshes with their spectral embeddings differ by a rotation and some stretching.



(b) A human mesh and its stretched version. Their spectral embeddings differ by a rotation.

Figure 4. Stretching in spatial domain induces rotation and non-rigid transformations in the spectral domain.

lieve that such transformations in the spectral domain, as well as eigenmode switchings, are the result of non-uniform stretching in the shapes. Obviously, a matching algorithm must be able to deal with all these transformations in order to operate robustly.

5. Our algorithm

Observe that iterative alignment techniques, e.g., [6], can work quite well when the initial shapes are approximately aligned, while spectral embedding can automatically remove the effects of rigid-body transformations, uniform scaling, and shape bending. Hence, a natural approach would be to perform non-rigid alignment in the spectral domain before computing the matching. The only obstacle now is to handle reflections caused by eigenvector switching and sign flips, as they can introduce large discrepancies into the initial configurations of the shapes to be matched. An exhaustive solution would be to consider all possible eigenvector orderings and sign flips and choose the one that gives the best matching. However, if k eigenvectors are used for embedding, there are $k! \times 2^k$ possibilities to exhaust; this is quite expensive even for small values of k.

We can use a simple, greedy heuristic. Let us first consider a very low dimensional embedding, e.g., with only two eigenvectors. We exhaustively find the best possible ordering and signs of these few eigenvectors. Now we incrementally add one eigenvector at a time and at each step, compute the best possible position and sign of the new eigenvector. This results in $O(k^2)$ possibilities to compare, greatly reducing the time complexity. Due to the rapid decay of eigenvalues and eigenvalue scaling, we never find it necessary to use more than $k = 6$ eigenvectors to arrive at a satisfactory mesh correspondence. So k is always small.

Once we have aligned the two spectral embeddings properly, we attempt to transform one embedding into another. Due to the presence of non-rigid deformations in the spectral domain, we modify the original rigid ICP algorithm [4] by replacing its transformation model with the use of thin-plate splines. Thin-plate splines are well-known and have been applied to model non-rigid transformations before [3, 6] in the context of 2D shape registration. A brief overview of this technique is given in the Appendix.

Now consider two 3D meshes M_1 and M_2 with n_{M_1} and n_{M_2} vertices, respectively. Without loss of generality, assume that $n_{M_1} \leq n_{M_2}$. Let us describe each step of our spectral correspondence algorithm in details below.

1. **Geodesic affinities:** Construct Gaussian affinity matrix A where

$$A_{ij} = e^{\frac{-d_{ij}^2}{2\sigma_{M_1}^2}}$$

   where d_{ij} is the geodesic distance between vertex i and j in M_1. The Gaussian kernel width σ_{M_1} is set to be the maximum geodesic distance between any two vertices in M_1. The performance of our method is relatively invariant to the choice of σ_{M_1} as long as it is set to a sufficiently large value. Similarly, we construct B, the Gaussian affinity matrix for mesh M_2.

2. **Spectral embeddings:** The affinity matrices A and B are eigenvalue decomposed and the resulting spectrum are truncated to k. Each of the k eigenvectors is scaled with the square root of its corresponding eigenvalue. These steps have already been described in details in Section 4.1 and 4.2.

   Note that if the Gaussian width is sufficiently large, the row-sums of the affinity matrix are almost constant. As a result, the first eigenvector of the matrix will be close to a constant vector and can be safely ignored. From now on, we denote by $\hat{A} \in \mathbf{R}^{n_{M_1} \times (k-1)}$ and $\hat{B} \in \mathbf{R}^{n_{M_2} \times (k-1)}$, as first defined in Equation 2, the $(k-1)$-dimensional embeddings of M_1 and M_2, respectively, where the first eigenvector is disregarded. $\hat{A}$ and $\hat{B}$ are essentially $n_{M_1} \times (k-1)$ and $n_{M_2} \times (k-1)$ matrices where the i^{th} rows of $\hat{A}$ and $\hat{B}$ are the $(k-1)$-dimensional spectral embedding coordinates of the i^{th} vertices of meshes M_1 and M_2 respectively. In all our

experiments, we have used $k = 5$ or 6 hence giving a 4 or 5-dimensional spectral embedding after disregarding the first eigenvector.

3. **Eigenvector reordering and sign correction**: We keep the ordering and signs of the eigenvectors of one mesh, e.g., M_1, fixed. With either the exhaustive search or the greedy heuristic, we need to compute the cost of a correspondence, which we describe below. First, we obtain a best matching C based simply on the L_2 metric; other metric, such as the Chi-square or Mahalanobis distance is also possible. Specifically, for a vertex $v_i^{M_1}$ of mesh M_1, $v_{C(i)}^{M_2}$ is the corresponding vertex of M_2, where

$$C(i) = \text{argmin}_j \|\hat{A}_i - \hat{B}_j\|. \quad (3)$$

Here $\hat{A}_i$ and $\hat{B}_j$ denote the spectral embedding coordinates of vertex i in mesh M_1 and vertex j in mesh M_2 respectively (i.e. the i^{th} and the j^{th} row of the matrices $\hat{A}$ and $\hat{B}$ respectively). The cost of the correspondence C is given by the sum:

$$cost(C) = \sum_{i=1}^{n_{M_1}} \|\hat{A}_i - \hat{B}_{C(i)}\|$$

We choose the ordering and signs of the eigenvectors for mesh M_2 which give the minimum $cost(C)$.

4. **Non-rigid alignment:** Once the eigenvectors for two shapes have consistent ordering and signs, we perform a non-rigid alignment using ICP modified by thin-plate splines (Refer to the Appendix). The pseudo-code for this alignment procedure is given below.

   Given two spectral embeddings $\hat{A}$ and $\hat{B}$,

   (a) Initialize parameters d, w, λ.

   (b) Transform $\hat{B}$ into $\hat{\hat{B}}$ using the transformation parameters d and w.

   (c) Update correspondence C using Equation (3) after replacing $\hat{B}_j$ with $\hat{\hat{B}}_j$.

   (d) Given the correspondence C, update transformation parameters using Equation (5).

   (e) Update the regularization parameter λ.

   (f) Repeat from Step 2 until convergence.

   We have found experimentally that 5 to 10 iterations of the iterative alignment are sufficient to align the embeddings. The value of the regularization parameter λ is set to be the mean distance between all embedded point pairs. As shown in [3], this scale-dependent assignment of λ is robust to scaling of the point sets.

5. **Proximity-aided matching:** For dense correspondence it is hard to distinguish between near-by points using an alignment and correspondence procedure based on optimizing a global energy, which is the case in our approach. In order to improve correspondence locally, we perform matching using a heuristic based on point proximity. Specifically, we first select a small number of *anchor point* pairs. These are point pairs that are best matched (that is, pairs contributing least to the correspondence cost), but that are also not too close to each other. Now for finding the correspondence cost between two points, we not only consider the L_2 distance between their (non-rigidly aligned) spectral embeddings, but also the difference between their geodesic proximities to these anchor points.

   The anchor point pairs are computed as follows. Consider the $(n_{M_1} \times n_{M_2})$ matrix Z of correspondence costs between all points of M_1 and M_2. That is,

$$Z_{ij} = \|\hat{A}_i - \hat{B}_j\|.$$

   The first anchor point pair $(a_1^{(1)}, a_2^{(1)})$, where $a_1^{(1)}$ is a vertex of M_1 and $a_2^{(1)}$ is a vertex of M_2, is selected as the pair with least correspondence cost. That is,

$$(a_1^{(1)}, a_2^{(1)}) = \text{argmin}_{(i,j)}(Z_{ij}).$$

   The second anchor point pair is calculated in the same way. However, we would need the anchor points to be far from each other over the mesh. Hence, before finding the second pair, we modify the matrix Z so that points close to the first anchor point are penalized. The new correspondence cost matrix is given by:

$$Z_{ij}^{(1)} = Z_{ij} - \frac{1}{2}[dist^{M_1}(i, a_1^{(1)}) + dist^{M_2}(j, a_2^{(1)})]$$

   where $dist^M(i, j)$ is the geodesic distance between the i^{th} and j^{th} vertex of mesh M. Now, the second anchor point pair is given by:

$$(a_1^{(2)}, a_2^{(2)}) = \text{argmin}_{(i,j)}(Z_{ij}^{(1)}).$$

   This process can be repeated to obtain more anchor point pairs. With the anchor points, we modify Equation (3) for finding the best correspondence C to incorporate the proximity cost:

$$C(i) = \text{argmin}_j \Big[\|\hat{A}_i - \hat{B}_j\| \sum_{l=1}^{h} \alpha_l \cdot \|dist^{M_1}(i, a_1^{(l)}) - dist^{M_2}(j, a_2^{(l)})\| \Big]$$

   where h is the number of anchor point pairs chosen and the α_l's are free parameters set by the user.

The success of the proximity heuristic depends on two factors: quality of the anchor point pairs and geodesic distances. Since the meshes are already well aligned, choosing the anchor point pairs to be the most trusted matches are expected to be robust. However, the dependence on geodesic distances may cause sensitivity of the heuristic to stretching in the shapes. Hence, fixing a large number of anchor point pairs can render the matching non-robust. Thus we restrict to fixing only three anchor pairs and set $\alpha_1 = \alpha_2 = \alpha_3 = 1$ for all our experiments.

6. Experimental results

To evaluate a mesh correspondence algorithm, we handpick a small number (17 to 20) of feature points on both of the meshes to be matched, where the number of vertices in these meshes vary between 100 to 300. The ground-truth correspondence between the features is determined by human. Now we compute a correspondence using the algorithm and record the percentage of correct correspondences obtained. Figure 6 shows a comparison between our algorithm and other well-known schemes, on eight test cases. The shapes to be matched in the test cases are shown in Figure 7; each pair of shapes exhibit some degree of non-rigid deformations. We now describe briefly the other schemes we have experimented with and our experimental setup.

- **TPS-RPM [6] in spatial domain:** This is one of the most successful non-rigid ICP algorithms. It combines thin-plate splines, soft assign, and deterministic annealing to achieve robust correspondence. But as we have shown in Figure 1, it is susceptible to poor initial alignment, such as a difference in rotation.

  To improve its performance, we *manually* and rigidly align the two shapes to be matched, attempting to neutralize any rotation or translation between them; this is done for all the three spatial-domain schemes we have experimented with. However, bending in the shapes still cause the algorithm to perform poorly.

- **Robust ICP [35] in spatial domain:** This method is a recent variant of the original ICP [4] algorithm. It uses a hierarchical approach to achieve robust registration of 3D point sets. We use it as a representative of the rigid iterative alignment schemes.

- **Shape context [3] in spatial domain:** We use a trivial 3D extension of the original 2D shape context of Belongie et al. [3] as a representative correspondence scheme based on local shape descriptors. Shape context is one of the most successful local descriptors for image analysis [25].

- **Shapiro and Brady [31]:** This is one of the early and best-known spectral point correspondence algorithms. It uses L_2 distance to compute a best matching, using their greedy sign correction but with no eigenvector reordering or eigenvalue scaling of eigenvectors.

- **Improved Shapiro and Brady:** Only eigenvalue scaling is incorporated into the original algorithm.

In each test case, $k = 6$ eigenvectors are used. Using more eigenvectors does not change the result due to eigenvalue scaling. Three out of the eight cases, Armadillo-Human, the Hands-1, and the Hands-3, have eigenvector switching occurring. In six of the eight cases, the greedy heuristic for eigenvector reordering and sign correction is successful; we shall provide a remark on this issue in the next section. The results shown are obtained by the expensive exhaustive search. Hence all results are limited to meshes with a few hundred vertices. In all test cases, no more than 10 iterations of our non-rigid ICP procedure are needed. In several cases, the procedure converges in less than 5 iterations. In terms of results, as can be seen from Figure 6, our algorithm clearly outperforms the state-of-the-art correspondence schemes mentioned above.

In Figure 7, we show some matching results obtained from our algorithm. The matching is shown by coloring the vertices of the meshes in an appropriate way. We first assign colors to the vertices of one of the two meshes, e.g., M_2. Then the color for the i^{th} vertex of mesh M_1 is set to be the color of the $C(i)^{th}$ vertex of M_2, where C is the correspondence found by our algorithm. This way, a good correspondence will induce a coloring that is consistent on both meshes. To show the meaningfulness of the correspondence obtained, we carefully assign different colors to different parts of the mesh M_2. Clearly, our algorithm matches bent shapes well, as well, it behaves robustly against moderate stretching in the shapes, e.g., the Armadillo vs. the human and the lion vs. the horse.

Note that in the Human:Human and Lion:Horse pairs, in Figure 7, which are of symmetric shapes, the correspondence is symmetrically switched. Namely, the right hand of one human is matched to the left hand of the other, etc. Similarly, the right leg of the lion is matched to the left leg of the horse, etc. This occurs since we define affinities based on geodesic distance, which is an intrinsic measure that cannot distinguish between symmetric points. As such, the left hand and the right hand of the human are equally good matches for the right hand of the other human. By chance, the Armadillo:Human, Cow:Lion, and Airplanes pairs return the right correspondences.

One solution to the above symmetry problem would be to carefully select the right sign of the eigenvectors. For shapes where there is one plane of symmetry, there will be two possible sign configurations of the eigenvectors that

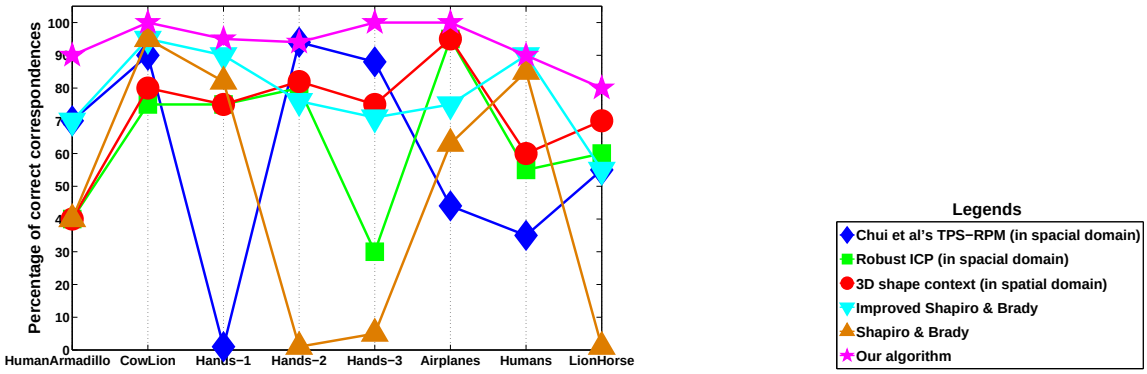


Figure 5. A comparison between several correspondence algorithms, including ours. The percentage of correct correspondences is plotted.

would give the minimum correspondence cost. These can be detected and the right configuration can be picked by inspecting it visually. As the number of symmetry planes increase, more sign configurations will give the minimum correspondence cost. A more analytical solution would be to define affinities in a symmetry-distinguishing way.

Figure 8 gives additional correspondence results obtained using our algorithm on numerous articulated shapes. In each of shape class, one per row, all the shapes are matched to a single reference shape (the first shape in each row of 8) and correspondence obtained is color coded in accordance with the colors on the reference shape. Apart from showing the effectiveness of our method, e.g., see the second row of Figure 8, these examples also reveal some of its limitations which we discuss below:

1. **Effect of intrinsic shape symmetry:** As explained earlier, due to the intrinsic nature of the affinity matrix, our method is not guaranteed to match symmetric shapes correctly, as shown in Figure 8(1-c) and Figure 8(4-c) and (4-d). In all three cases, the sign configuration of the eigenvectors that gives the lowest correspondence cost leads to counterintuitive correspondence results. Our method succeeds in all the remaining cases in row 1, 2, and 4, although in each case, the next best eigenvector sign configuration, which has a correspondence cost extremely close to the lowest cost, would give a symmetrically flipped matching. This shows that the correspondence might very well have been symmetrically flipped in these cases too. Note that the correspondence may be symmetrically flipped, but it is nevertheless still consistent across the shape.

2. **Effect of topological changes:** Since our method largely depends on geodesic distances, topological changes can seriously harm the correspondence computation. This effect is visible in Figure 8(3-e) and (3-f), where the fingers of the hands are connected to the palm which would change the connectivity of the mesh, as well as the geodesic distances, drastically, resulting in unnatural correspondence results. Correct recovery of the correspondence between the fingers in this case appears to be a rather difficult problem, without some level of *a prior* knowledge.

3. **Unreliable geodesic distances:** Figure 8(4-e) shows a bird shape that is very similar to the reference figure for this group, Figure 8(4-a). However, the correspondence obtained is incorrect. We suspect that this is mainly due to the unreliability of geodesic distances on the wings of the bird that contains many "cuts". Hence, even though the shapes look similar in the spatial domain, their embeddings are rather different.

4. **Non-robustness of L_2 cost for exhaustive search:** Close inspection of Figure 8(1-b) reveals that the correspondence obtained is inconsistent: the left arm is colored orange which means that the left leg must be colored blue which is not the case (note that this is different from the symmetry issue discussed above). This should not have been the case as the shapes are topologically sound and the geodesic distances are computed robustly. The problem becomes clear when we examine the result of the exhaustive reordering of eigenvectors. It turns out that for this shape, the exhaustive reordering does not give the right ordering of the eigenvectors, as shown in Figure 9. After further investigation we find that that the problem lies with the crude L_2 cost measure used in arriving at the reordering. A more robust cost measure should be sought.

7. Conclusion and future work

In this paper, we present a hybrid approach to finding a one-to-one correspondence between the vertices of two 3D meshes. We first transform the meshes into the spectral domain, based on geodesic affinities, and then match the spectral embeddings after taking appropriate steps to ensure a consistent ordering and sign assignment of the eigenvectors. Eigenvalue scaling of the eigenvectors renders our algorithm robust against difference in mesh sizes and choice of the dimensionality of the embeddings. Our method does not need a pre-selected set of feature vertices and can be completely automated. It is invariant against rigid transformations, uniform scaling, and shape bending. Experimentally, we find it to be robust against moderate stretching in the shapes as well, relying on thin-plate splines for non-rigid alignment in the spectral domain, and it outperforms well-known existing shape correspondence schemes.

The time complexity for computing the spectral embeddings and the correspondence cost, provided that an ordering and signs of the eigenvectors have been determined, is $O(n^2 \log n)$, where n is the number of vertices in the larger mesh. Note that this is inherent to the spectral approach, since the first step, which computes the pair-wise affinities, already requires $O(n^2 \log n)$ time. One effective remedy is to use sub-sampling techniques and extrapolation, e.g., Nyström method [11], to obtain approximate spectral embeddings. This reduces the cost of computing the spectral embeddings to $O(pn \log n)$, where p is the number of samples chosen. In our subsequent work [24], we demonstrate that excellent correspondence results can be obtained with only $p = 10$ samples for meshes with hundreds of vertices. To reduce the cost of extracting correspondences, where the naïve best matching would require $O(n^2)$ time, we can take advantage of the accurate alignments that have already been obtained and apply spatial partitioning to speed up the search for correspondence pairs.

The other limitation of our current approach, in terms of computational cost, is its reliance on an exhaustive search to find a consistent eigenvector ordering and sign assignment. The greedy reordering approach is fast but it does not always give the correct result. Analytically, the problem of finding a reordering and sign assignment which would lead to the best correspondence, e.g., according to the simple L_2 distance, is as hard as the graph isomorphism problem. We would like to look into fast approximation algorithms for this problem and adopt it for our purpose.

Quality-wise, an important issue is related to the quality of the correspondence cost used in determining the eigenvector ordering and sign assignment. Currently, we are using L_2 and in some rare cases as shown earlier, even the exhaustive search would return a poor eigenvector ordering or sign assignment. This shows that a better cost function is still required. In addition, we plan to address the other limitations of our current method, including the handling of shape symmetry and non-manifold meshes.

Finally, we would like to investigate possible definitions of the point affinities that are robust, if not invariant, to stretching within perceptually salient parts of a shape. This would offer an alternative to using non-rigid alignment in the spectral domain and avoid having to find a consistent eigenvector ordering or sign assignment.

References

[1] M. Alexa, "Recent Advances in Mesh Morphing," *Computer Graphics Forum,* Vol. 21, No. 2, pp. 173-196, 2002.

[2] D. Anguelov, P. Srinivasan, D. Koller, S. Thrun, H. Pang and J. Davis, "The Correlated Correspondence Algorithm for Unsupervised Registration of Nonrigid Surfaces," *NIPS,* 2004.

[3] S. Belongie, J. Malik, and J. Puzicha, "Shape Matching and Object Recognition Using Shape Contexts," *IEEE Trans. on PAMI,* Vol. 24, No. 24, pp. 509-523, 2002.

[4] P. J. Besl and N. D. Mckay, "A Method for Registration of 3-D Shapes," *IEEE Trans. on PAMI,* Vol. 14, No. 2, pp. 239-256, 1992.

[5] T. Caelli and S. Kosinov, "An Eigenspace Projection Clustering Method for Inexact Graph Matching," *IEEE Trans. on PAMI,* Vol. 26, No. 4, pp. 515-519, 2004.

[6] H. Chui and A. Rangarajan, "A new point matching algorithm for non-rigid registration," *Computer Vision and Image Understanding,* 89:114-141, 2003

[7] M. Carcassoni and E. R. Hancock, "Spectral Correspondence for Point Pattern Matching," *Pattern Recognition,* Vol. 36, pp. 193-204, 2003.

[8] M. Carcassoni and E. R. Hancock, "Correspondence Matching with Modal Clusters," *IEEE Trans. on PAMI,* Vol. 25, No. 12, pp. 1609-1615, 2003.

[9] C. Eckart and G. Young, "The Approximation of One Matrix by Another of Lower Rank," *Psychometrika,* Vol. 1, pp. 211-218, 1936.

[10] A. Elad and R. Kimmel, "On Bending Invariant Signatures for Surfaces," *IEEE Trans. on PAMI,* Vol. 25, No. 10, pp. 1285-1295, 2003.

[11] C. Fowlkes, S. Belongie, F. Chung, and J. Malik, "Spectral Grouping Using the Nyström Method," *IEEE Trans. on PAMI,* Vol. 26, No. 2, pp. 214-225, 2004.

[12] M. Garland, *QSlim Simplification Software — qslim 2.1,* http://graphics.cs.uiuc.edu/ garland/software/qslim.html.

[13] T. Gatzke, C. Grimm, M. Garland and S. Zelinka, "Curvature Maps for Local Shape Comparison," in *Proc. Shape Modeling International*, 2005.

[14] N. Gelfand, N. Mitra, L. Guibas, H. Pottmann, "Robust Global Registration," *SGP,* 2005.

[15] G. H. Golub and C. F. Van Loan, *Matrix Computations,* John Hopkins University Press, 1996.

[16] C. Gotsman, X. Gu, and A. Sheffer, "Fundamentals of Spherical Parameterization for 3D meshes," *ACM Transaction on Graphics,* Vol. 22, No. 3, pp. 358-363, 2003.

[17] A. Johnson and M. Hebert, "Using Spin-Images for Efficient Multiple Model Recognition in Cluttered 3-D Scenes," *IEEE Trans. on PAMI*, pp. 433-449, 1999.

[18] M. Körtgen, G-J. Park, M. Novotni, R. Klein, "3D Shape Matching with 3D Shape Contexts," *Seventh Central European Seminar on Computer Graphics,* 2003.

[19] R. Kolluri, J. R. Shewchuk, and J. F. O'Brien, "Spectral surface reconstruction from noisy point clouds," *Proc. of the 2004 Eurographics/ACM SIGGRAPH symposium on Geometry processing,* pp. 11-21, 2004.

[20] V. Kraevoy, A. Sheffer, and C. Gotsman, "Matchmaker: Constructing Constrained Texture Maps," *ACM SIGGRAPH,* pp. 326-333, 2003.

[21] V. Kraevoy and A. Sheffer, "Cross-Parameterization and Compatible Remeshing of 3D Models," *ACM SIGGRAPH,* 2004.

[22] R. Liu and H. Zhang, "Segmentation of 3D Meshes through Spectral Clustering," *Proc. of Pacific Graphics,* pp. 298-305, 2004.

[23] X. Li, I. Guskov, "Multiscale Features for Approximate Alignment of Point-based Surfaces," *SGP,* 2005.

[24] R. Liu, V. Jain, and H. Zhang, "Subsampling for Efficient Spectral Mesh Processing," *Computer Graphics International* June, 2006 (to appear).

[25] K. Mikolajczyk and C. Schimid, "A Performance Evaluation of Local Descriptors," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* Vol. 27, No. 10, pp. 1615-1630, 2005.

[26] A. Y. Ng, M. I. Jordan, Y. Weiss, "On Spectral Clustering: Analysis and An Algorithm," in *Advances in Neural Information Processing Systems* 14, pp. 857-864, 2002.

[27] E. Praun, W. Sweldens, and P. Schröder, "Consistent Mesh Parameterizations," *ACM SIGGRAPH,* pp. 179-184, 2001.

[28] S. Rusinkiewicz and M. Levoy, "Efficient Variants of the ICP Algorithm," In *The Third International Conference on 3-D Digital Imaging and Modeling,* pp. 145-152, 2001.

[29] J. Schreiner, A. Asirvatham, E. Praun, and H. Hoppe, "Inter-Surface mapping," *ACM Trans. Graph.,*, Vol. 23, No. 3, pp. 870-877, 2004.

[30] G. Scott and H. Longuet-Higgins, "An Algorithm for Associating the Features of Two Patterns," *Proc. Royal Soc. London,* Vol. B244, 1991.

[31] L. S. Shapiro and J. M. Brady, "Feature Based Correspondence: An Eigenvector Approach," *Image and Vision Computing,* Vol. 10, No. 5, pp. 283-288, 1992.

[32] R. Sumner and J. Popovic, "Deformation Transfer for Triangle Meshes," *ACM Transactions on Graphics,* Vol. 23(3), 2004.

[33] S. Umeyama, "An Eigen Decomposition Approach to Weighted Graph Matching Problems," *IEEE Trans. on PAMI,* Vol. 10, pp. 695-703, 1988.

[34] R. Zayer, C. Rössl, Z. Karni and H.-P. Seidel, "Harmonic Guidance for Surface Deformation," *Computer Graphics Forum (Proceedings of Eurographics),* Vol. 24(3), pages 601-609, 2005.

[35] T. Zinber, J. Schmidt and H. Niemann, "A Refined ICP Algorithm For Robust 3D Correspondence Estimation," in *Proc. of International Conference on Image Processing*, 2003.

Appendix: Thin-plate splines

The thin plate spline is a generalization of cubic splines to higher dimensions and it contains affine transformations as a special case. With non-rigid transformations, there are infinitely many ways of transforming a point set into another. Thin plate splines are effective because of their smoothness constraints which discourage arbitrary mappings. In the limit of this smoothness constraint the thin plate spline model reduces to an affine transformation model. The thin plate spline transformation function $f : x \in \mathbf{R}^k \rightarrow y \in \mathbf{R}^k$ maps a point set $X = \{x_1, x_2, \ldots, x_n\}$ in k (say $k = 2$) dimensional space to another point set $Y = \{y_1, y_2, \ldots, y_n\}$ by minimizing the following energy function:

$$E(f) = \sum_{i=1}^{n} \|y_i - f(x_i)\| + \lambda \int\int \left[(\frac{\partial^2 f}{\partial x^2})^2 + 2(\frac{\partial^2 f}{\partial x \partial y})^2 + (\frac{\partial^2 f}{\partial y^2})^2 \right] dxdy \quad (4)$$

where λ is the regularization (smoothing) parameter. Note that the correspondence between X and Y is assumed to be given. Hence, point y_i is the matching point for x_i. The unique f that minimizes the above energy function has the form:

$$f(x_i, d, w) = x_i \cdot d + \phi(x_i) \cdot w,$$

where x_i is now in $(k+1)$-dimensional homogeneous coordinates, d is a $(k+1) \times (k+1)$ affine transformation matrix, w is an $n \times (k+1)$ warping coefficients matrix and $\Phi(x_i)$ is a vector of length n such that $\phi_j(x_i) = -\|x_j - x_i\|$.

As shown in [3], the transformation (d, w) that minimizes the energy can be calculated by solving the following system:

$$\begin{bmatrix} K & X^T \\ X & 0 \end{bmatrix} \begin{bmatrix} w \\ d \end{bmatrix} = \begin{bmatrix} Y \\ 0 \end{bmatrix} \quad (5)$$

Here, K is the matrix $(\Phi - \lambda I)$ where, I is an identity matrix of appropriate size and Φ is an $(n \times n)$ matrix whose i^{th} row is $\phi(x_i)$, that is, $\Phi_{ij} = -\|x_j - x_i\|$.

Using these transformation parameters, we transform the point set X to point set Y and then recompute the correspondence. This process is iterated until convergence.

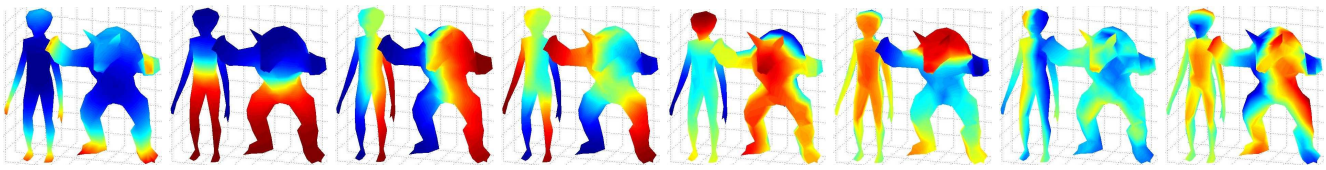


Figure 6. Eigenvector plots for two shapes, both with 252 vertices. The first 8 eigenvalues are [205.6, 11.4, 4.7, 3.8, 1.8, 0.4, 0.26, 0.1] and [201.9, 10.9, 6.3, 3.4, 1.8, 1.2, 0.31, 0.25], respectively.

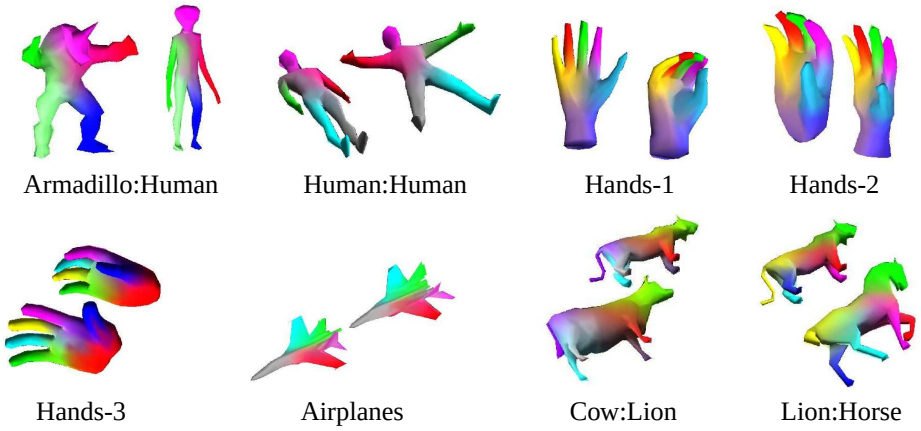


Figure 7. Correspondence results obtained from our algorithm, shown with color plots.

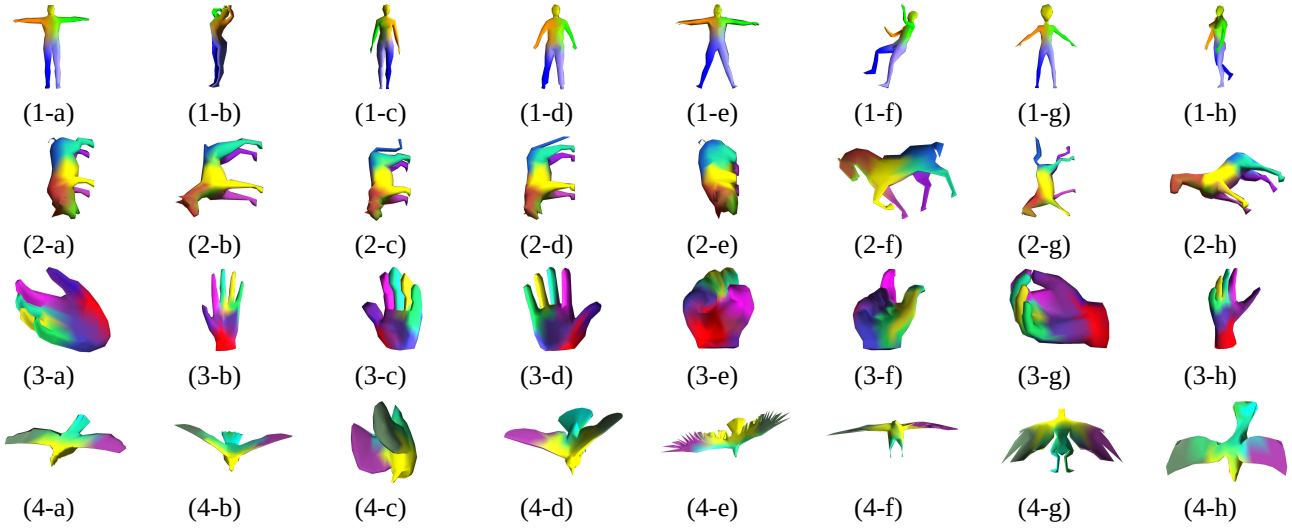


Figure 8. Row 1: correspondence results for human shapes. Row 2: for animal shapes. Row 3: for hand shapes. Row 4: for bird shapes.

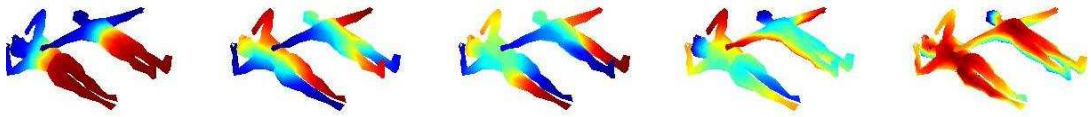


Figure 9. Incorrect eigenvector ordering is obtained even after exhaustively reordering the eigenvectors for shapes in Figure 8(1-a) and (1-b).