

Fast Multiple Shape Correspondence by Pre-Organizing Shape Instances

Brent C. Munsell, Andrew Temlyakov and Song Wang

Department of Computer Science and Engineering
University of South Carolina, Columbia, SC 29208, USA

munsell@engr.sc.edu, temlyaka@engr.sc.edu, songwang@engr.sc.edu

Abstract

Accurately identifying corresponded landmarks from a population of shape instances is the major challenge in constructing statistical shape models. In general, shape-correspondence methods can be grouped into one of two categories: global methods and pair-wise methods. In this paper, we develop a new method that attempts to address the limitations of both the global and pair-wise methods. In particular, we reorganize the input population into a tree structure that incorporates global information about the population of shape instances, where each node in the tree represents a shape instance and each edge connects two very similar shape instances. Using this organized tree, neighboring shape instances can be corresponded efficiently and accurately by a pair-wise method. In the experiments, we evaluate the proposed method and compare its performance to five available shape correspondence methods and show the proposed method achieves the accuracy of a global method with speed of a pair-wise method.

1. Introduction

Accurately identifying corresponded landmarks from a population of shape instances is the major challenge in constructing statistical shape models. In constructing a statistical shape model the first step is to identify a set of landmarks along the contour of each shape instance in the population, where a *landmark* is defined as a point of correspondence across the set of shape instances in order to examine and measure shape change [1][2]. It is critical that the identified landmarks be of the same number, well corresponded, and sufficiently dense to represent the underlying continuous contour of each shape instance. The constructed statistical shape model can then be applied to address many important applications in computer vision and medical image analysis. For example, in [5][15] statistical shape models are successfully used to guide image segmentation by detecting structures with desirable shapes. In [1][3], statistical shape models are used to accurately locate the subtle differ-

ence of the corpus-callosum shapes between schizophrenia patients and normal controls.

Developing more accurate and efficient shape correspondence methods has been widely investigated in the past several years. Previous shape correspondence methods are developed by optimizing some assumed physical or mathematical model. For example, the minimum description length (MDL) [8][22] shape correspondence method considers the landmarks of each shape instance to be well corresponded when the bit-length required to transmit the constructed point distribution model (PDM) [5] is minimized. In [13], a binary tree is used to determine the initial estimate of each shape instance's landmarks, then using this initial estimate shape correspondence is achieved when the landmarks lead to a more compact PDM. In [1] [18] [23], 2D thin-plate splines are used to model the nonrigid deformation between two shape instances and their identified landmarks are considered to be well corresponded when the thin-plate bending energy is minimized. In [16], spherical harmonics are used to model the difference between two shape instances, and shape correspondence is achieved when the spherical harmonic coefficients that minimize the differences in curvature, distance, and normal direction between the identified landmarks are found. In [24], the medial-axis is used to identify matched contour segments between two shape instances, the landmarks within these identified segments are considered to be well corresponded when an optimal one-to-one point matching cost is found.

In general, the shape correspondence methods briefly described above can be grouped into one of two categories: global methods and pair-wise methods. For global methods, an objective function which considers the entire population of shape instances is optimized. For pair-wise methods, one shape instance from the population is designated as the template and the remaining target shape instances are optimized to the template one by one. While global methods may produce a more accurate shape correspondence they tend to scale poorly when the population size becomes very large. Genetic [8], and gradient descent [12] techniques have been used to address this concern, however these optimization

methods may require a large number of iterations to converge and even if convergence is achieved there is no guarantee it will converge to some global optimum. On the other hand, since a pair-wise method only considers two shape instances at any time, they tend to be less compute intensive and scale favorably to the size of the population. However, because a single template shape instance is chosen from the population pair-wise methods tend to be less accurate and can perform unsatisfactorily when the population has a large amount of variance [17].

In this paper we attempt to address the limitations of global and pair-wise methods by developing a shape correspondence method that pre-organizes the population of shape instances in a tree. Specifically, this is achieved by constructing a minimum spanning tree (MST), where each node represents a shape instance and each edge connects two very similar shape instances. The pre-organization step provided by the MST allows us to incorporate global information about the population of shape instances prior to shape correspondence. A root node is then selected which represents the starting shape instance, and then using the constructed MST and the selected root node neighboring shape instances can be corresponded efficiently and accurately using a pair-wise method.

The remainder of this paper is organized as follows: in Section 2, we describe in detail the proposed shape correspondence method. In Section 3, we evaluate the performance of the proposed shape correspondence against five available shape-correspondence methods. In Section 4, we provide a brief conclusion.

2. The Proposed Method

2.1. Problem Formulation

In this section we develop a new shape correspondence method that attempts to address the limitations of global and pair-wise methods. We achieve this by pre-organizing the population of shape instances by constructing a minimum spanning tree (MST), where each node in the MST represents a shape instance and each edge connects two very similar shape instances. A root node that represents the starting template shape instance is selected. Then using the constructed MST, and the selected root node, we walk the MST in a depth-first fashion corresponding neighboring shape instances using a pair-wise method, where the current node represents the template shape instance and it's child nodes are the target shape instances.

Unlike traditional pair-wise methods that choose a single template shape instance from the population, the proposed method defines $|V| - |V_t|$ template shape instances, where $|V|$ is the number of nodes and $|V_t|$ is the number of leaf nodes (nodes with no children) in the MST. The landmarks of the template shape instance are defined by the path of

edges from the root node to the leaf node, where each successive edge slightly adjusts the landmarks defined by the previous template. We illustrate this process using the example shapes shown in Fig. 1, where the first shape instance is designated the template and the landmarks of the second shape instance are corresponded to those of the first. When completed, the second shape instance is designated the template and the landmarks of the third shape instance are corresponded to those of the second.

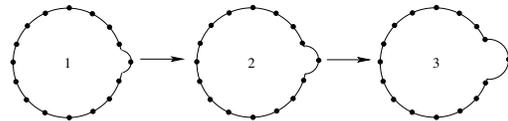


Figure 1. Three example shape instances where the first shape instance is designated the template and the landmarks of the second shape instance are corresponded to those of the first. When completed the second shape instance is designated the template and the landmarks of the third shape instance are corresponded to those of the second.

Because the proposed method only corresponds neighboring shape instances that are very similar, it is reasonable to assume that the error introduced by the process illustrated in Fig. 1 may be less than directly corresponding two shape instances whose difference is quite large, e.g. designating the first shape instance as the template, and the landmarks of the third are corresponded to those of the first. Furthermore, if a large number of shape instances is required to well represent the variance in the population, the proposed method scales favorably in such conditions without greatly impacting the speed of the method. By increasing the size of the population, the MST can better estimate the similarity between neighboring shape instances, resulting in a more organized MST which can lead to a more accurate shape correspondence.

This formulation brings us to several key issues which need to be addressed: (a) how to construct the MST, (b) how to determine the root node of the MST, (c) the algorithm used to correspond neighboring shape instances in the MST, and (d) how to determine and handle error conditions during shape correspondence.

2.2. Minimum Spanning Tree

Given a set of 2D shape instances $\tilde{S} = \{\tilde{S}_i, i = 1, \dots, n\}$ that represent continuous shape contours, we sequentially identify a set of m landmarks $\tilde{U}_i = \{\tilde{\mathbf{u}}_{ik}, k = 1, \dots, m\}$ along the boundary of each shape contour at equal arc-length intervals¹, where $\tilde{\mathbf{u}}_{ik} = (x_{ik}, y_{ik})$ rep-

¹For closed contours, the first landmark for each shape instance in the population is found using an exhaustive search method based on the thin-plate spline [23]. This first landmark is considered an approximation that can be refined during shape correspondence. For open contours, the first landmark is one of the end-points.

resents the k th landmark along \tilde{S}_i . We then normalize $\tilde{U} = \{\tilde{U}_i, i = 1, \dots, n\}$ by removing the scaling, rotation, and translation transformations using Procrustes analysis [11]. Using this normalized set of shape instances $U = \{U_i, i = 1, \dots, n\}$, we then estimate the set of normalized continuous shape contours $S = \{S_i, i = 1, \dots, n\}$ by densely interpolating the m landmarks of U_i for $i = 1, \dots, n$ using the Catmull-Rom cubic spline [4].

We then construct a fully connected undirected graph defined by n nodes, where node v_i represents shape instance U_i and the edge (v_i, v_j) that connects nodes v_i and v_j is weighted by

$$w(v_i, v_j) = \begin{cases} \Delta(S_i, S_j) \times \Omega(U_i, U_j) & , i \neq j \\ \infty & , i = j \end{cases} \quad (1)$$

for $i, j = 1 \dots, n$.

The first term $\Delta(S_i, S_j)$ in Eq. (1), measures the difference between two closed continuous shape contours using the Jaccard coefficient

$$\Delta(S_i, S_j) = 1 - \frac{|R(S_i) \cap R(S_j)|}{|R(S_i) \cup R(S_j)|}, \quad (2)$$

where $R(S)$ is the region enclosed by contour S , and $|R|$ computes the area of the region R . For open shapes, the first and last landmark are tied together to enclose the region. The difference is limited to a value between $[0, 1]$, where zero indicates S_i is exactly the same as S_j .

Since the Jaccard coefficient measures the region coincidence between two closed contours, this measure may be insensitive to shape differences. To overcome such conditions, the second term $\Omega(U_i, U_j)$ in Eq.(1) further weights the Jaccard coefficient by measuring the non-rigid shape difference between U_i and U_j

$$\Omega(U_i, U_j) = \frac{\beta(U_i, U_j) + \beta(U_j, U_i)}{2},$$

where $\beta(U_i, U_j)$ and $\beta(U_j, U_i)$ represents the bending energy, or the amount of energy required to deform U_i to U_j , and U_j to U_i respectively based on the 2D thin-plate spline model [1].

Specifically, if $P = \{\mathbf{p}_k, k = 1, \dots, m\}$ defines the template shape instance, and $U = \{\mathbf{u}_k, k = 1, \dots, m\}$ defines the target shape instance, where $\mathbf{p}_k = (\hat{x}_k, \hat{y}_k)$ and $\mathbf{u}_k = (x_k, y_k)$. The thin-plate spline finds a mapping $\mathbf{f} = (f_x, f_y)$ from P to U such that $x_k = f_x(\mathbf{p}_k)$ and $y_k = f_y(\mathbf{p}_k)$. In quadratic form, the bending energy is represented by

$$\beta(P, U) = \mathbf{x}^T \mathbf{L} \mathbf{x} + \mathbf{y}^T \mathbf{L} \mathbf{y}, \quad (3)$$

where $\mathbf{x} = (x_1, \dots, x_m)^T$, and $\mathbf{y} = (y_1, \dots, y_m)^T$ are vectors that contain the x and y coordinates of the landmarks in U . The bending matrix \mathbf{L} is the $(m \times m)$ upper

left sub-matrix of

$$\begin{bmatrix} \mathbf{K} & \mathbf{D} \\ \mathbf{D}^T & \mathbf{0} \end{bmatrix}^{-1},$$

where $\mathbf{D} = (\mathbf{1}, \hat{\mathbf{x}}, \hat{\mathbf{y}})$ is a $(m \times 3)$ matrix defined by column vectors $\hat{\mathbf{x}} = (\hat{x}_1, \dots, \hat{x}_m)^T$ and $\hat{\mathbf{y}} = (\hat{y}_1, \dots, \hat{y}_m)^T$ that contain the x and y coordinates of the landmarks in P . Furthermore, \mathbf{K} is an $(m \times m)$ matrix, where $k_{ab} = \mathbf{K}(a, b) = \|\mathbf{p}_a - \mathbf{p}_b\|^2 \log \|\mathbf{p}_a - \mathbf{p}_b\|$ for $a, b = 1, \dots, m$.

Using Eq. (3), the bending energy for $\beta(U_i, U_j)$ and $\beta(U_j, U_i)$ are calculated and the non-rigid shape difference $\Omega(U_i, U_j)$ between shape instance U_i and U_j is equal to the average of the bending energy values. If this non-rigid shape difference is small, then the shape instances represented by U_i and U_j describe similar shapes.

Given this constructed graph, the minimum spanning tree $T = (V, E)$ is found using either Prim's or Kruskal's algorithm [6], where T is a connected, acyclic, undirected graph defined by the set of vertices V with size $|V| = n$, and edges E with size $|E| = n - 1$, whose total edge weight is minimum.

2.3. Selecting the Root Node of the MST

Since the root node defines the starting template shape instance, it is important to pick a node in the MST that does not represent an outlier shape instance. One approach is to select a node that has the most neighbors. Such a node may be less likely to represent an outlier shape instance. Based on this assumption, the number of neighboring nodes for each node in V can be calculated as $D = \{\delta(v_i), i = 1, \dots, n\}$, where $\delta(v_i)$ is the degree of node v_i . The values in D are sorted in descending order resulting in a set of nodes with $\delta \geq 2$.

However, simply selecting the node with the largest number of neighboring nodes as the root node of the MST may not be appropriate. For example, the selected node may represent an outlier shape instance that is similar to several other outlier shape instances. To address this concern we select a node that *balances* the MST. In the context of this paper, the MST is balanced when neighboring nodes near the selected root node represent shape instances that have greater similarity to each other than those near the leaf nodes.

The method used to determine the root node of the MST is outlined as follows. Assume we have a MST $T = (V, E)$, where the length of the edge as shown in Fig. 2 is proportional to it's weight. Additionally, let $\{v_1, v_2, v_3, v_4\}$ be the set of sorted nodes with degree $\delta \geq 2$. If v_1 is selected as the root node of the MST, as illustrated in Fig. 2(a), then its

cost is defined as,

$$\begin{aligned}
 C(v_1) = & h(v_1)(w(v_1, v_2) + w(v_1, v_3) + w(v_1, v_4)) \\
 & + h(v_2)(w(v_2, v_5) + w(v_2, v_6)) \\
 & + h(v_3)(w(v_3, v_7) + w(v_3, v_8)) \\
 & + h(v_4)(w(v_4, v_9) + w(v_4, v_{10})),
 \end{aligned}$$

where $h(\cdot)$ is the height of the node, which is equal to the number of edges on the longest simple downward path to a leaf node [6], and $w(\cdot)$ is the weight of the edge calculated in Eq. (1). Further inspection shows that each edge is additionally weighted by the height of the node placing more emphasis on edges located near the selected root node.

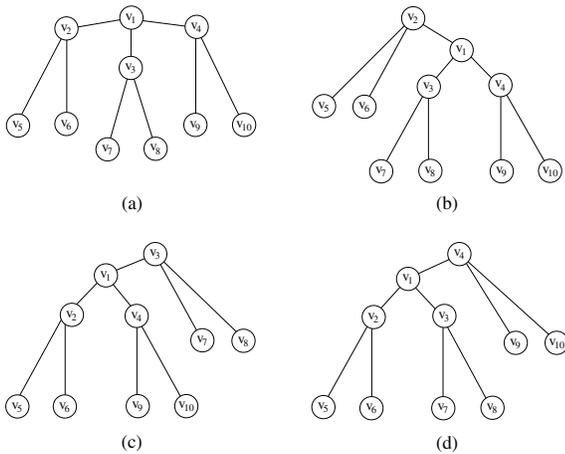


Figure 2. MSTs with node (a) v_1 selected as the root node, (b) v_2 selected as the root node, and (c) v_3 selected as the root node, and (d) v_4 selected as the root node.

In general, given a MST $T = (V, E)$, where $V = \{v_i, i = 1, \dots, n\}$ the cost of selecting node v_i as the root node of the MST is given by

$$C(v_i) = \sum_{k=1}^{|V|} \sum_{j=1}^{|\tilde{V}_k|} h(v_k)w(v_k, \tilde{v}_{kj}),$$

where $\tilde{V}_k = \{\tilde{v}_{kj}, j = 1, \dots, |\tilde{V}_k|\}$ is the set of nodes with parent v_k . For each node in V the cost is calculated, and the node with minimal cost is selected as the root node of the MST. The proposed cost favors rooted MST's such as those illustrated in Fig. 2(a), over those in Fig. 2(b) with v_2 as the root node, Fig. 2(c) with v_3 as the root node, and Fig. 2(d) with v_4 as the root node.

2.4. Proposed Algorithm

The next step is to correspond the landmarks for each shape instance using the rooted MST $T = (V, E)$ developed in Sections 2.2 and 2.3. The proposed algorithm is outlined in Algorithm 1. Given the MST T and the root node v as

input, on *line 1* the shape instance represented by node v is designated as the template shape instance P . *Line 2*, locates the child nodes of v , and the shape instance represented by child node \tilde{v} is designated as the target shape instance U on *line 3*.

Algorithm 1 CORRESPOND MST(T, v)

- 1: $P \leftarrow getTemplateShapeInstance(v)$
 - 2: **for all** $\tilde{v} \in children[v]$ **do**
 - 3: $U \leftarrow getTargetShapeInstance(\tilde{v})$
 - 4: $\min \phi(P, U)$
 - 5: CORRESPOND MST(T, \tilde{v})
 - 6: **end for**
-

On *line 4*, shape correspondence between the template shape instance and target shape instance is achieved by moving landmarks in the target shape instance to minimize the shape correspondence error [23]

$$\phi(P, U) = \beta(P, U) + \lambda R(U), \quad (4)$$

where $\beta(P, U)$ measures the bending energy between the target U and template P shape instances provided in Eq. (3), $R(U)$ defines the shape representation error of the target shape instance, and $\lambda > 0$ is a balance factor². Specifically, the shape representation error $R(U)$ measures the spatial distribution between the landmarks of the template and target shape instances and favors landmarks that are geometrically similar

$$\frac{l_k}{l_{k+1|m}} \approx \frac{\hat{l}_k}{\hat{l}_{k+1|m}},$$

where l_k is the curve length from landmark \mathbf{u}_k to $\mathbf{u}_{(k+1|m)}$ in the target shape instance, and \hat{l}_k is the curve length from landmark \mathbf{p}_k to $\mathbf{p}_{(k+1|m)}$ in the template shape instance. Based on the above, the representation error is calculated as follows

$$R(U) = \sum_{k=1}^m (l_k \hat{l}_{k+1|m} - \hat{l}_k l_{k+1|m})^2.$$

The shape correspondence error in Eq. (4) can be formulated as an quadratic programming problem which iteratively optimizes the sliding distance of the landmarks along the contour of the target shape instance. At each iteration the sliding distances are subject to several constraints such that no landmark slides beyond its neighbors thus preserving the topology of the target shape instance. Since the shape correspondence error in Eq. (4) only considers two

²This factor balances the bending energy and the shape representation error terms in Eq. (4) that prevents the degenerate case, where all the landmarks on the target shape move too close to each other.

shape instances, and is based on a local search algorithm, landmark sliding can produce a more accurate and reliable shape correspondence when the template and target shape instances are very similar.

Lastly, on *line 5* when the shape correspondence error has been minimized, or the maximum number of iterations has been reached, the shape instance represented by node \tilde{v} becomes the template shape instance and its child nodes become the target shape instances. Using this depth-first approach, the proposed algorithm recursively walks the MST until a leaf node is reached. Finally, the algorithm terminates when each node has been visited, i.e. the landmarks for each shape instance are corresponded.

2.5. Error Handling

During shape correspondence it is possible that the quadratic solver was unable to find the optimal sliding distances. In particular, in conditions when neighboring shape instances in the MST are not similar the quadratic solver may be unable to find a solution, or its progress towards convergence is unacceptably slow. In either case the proposed method attempts to handle these error conditions as follows.

Given the MST $T = (V, E)$, if the proposed method fails to minimize the shape correspondence error in Eq. (4) between the target shape instance represented by node \tilde{v} , and the template shape instance represented by node v , the method determines the size of the subtree $\tilde{T} = (\tilde{V}, \tilde{E})$ rooted at node \tilde{v} , where \tilde{T} is a subtree in T . If $|\tilde{V}| \ll |V|$, and the weight $w(v, \tilde{v})$ of edge (v, \tilde{v}) is significantly large, then it is reasonably safe to assume that the nodes in subtree \tilde{T} represent shape instances that may be outliers. In such conditions, the subtree \tilde{T} rooted at \tilde{v} is pruned from the MST and discarded.

However, if the number of nodes in $|\tilde{V}|$ is significantly large we need to develop a way to locate a new template shape instance in the MST for \tilde{v} . This is achieved by calculating the set of weights $w(v_k, \tilde{v})$ for $k = 1, \dots, |\tilde{V}|$ such that $v_k \neq v$ and $v_k \notin \tilde{V}$, where $w(\tilde{v}, v_k)$ is the weight of edge (\tilde{v}, v_k) provided in Eq. (1). The weights are then sorted in ascending order resulting a set of nodes $\{\hat{v}_1, \hat{v}_2, \dots, \hat{v}_z\}$ that represent candidate template shape instances for \tilde{v} , where \hat{v}_1 represents the node whose edge weight $w(\hat{v}_1, \tilde{v})$ is minimal. The previous edge (v, \tilde{v}) , as illustrated in Fig. 3(a), is broken and removed from the set of edges E , and a new edge (\hat{v}_1, \tilde{v}) , as illustrated in Fig. 3(b), is created and added to the set of edges in E .

The proposed method then attempts to minimize the shape correspondence error in Eq. (4), where the template shape instance is represented by node \hat{v}_1 . If failure occurs with \hat{v}_1 , the next candidate node \hat{v}_2 becomes the template shape instance. This process is repeated until the target shape instance is successfully corresponded, or when each

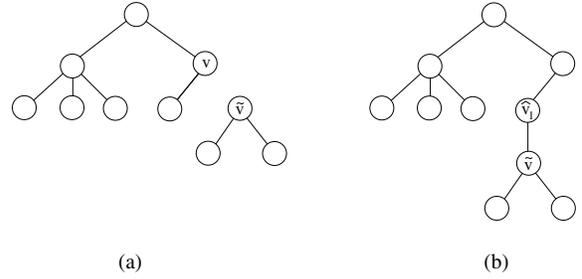


Figure 3. The proposed method fails to correspond the target shape instance to the template. (a) the edge that connects the two failed nodes is broken and removed from E , and (b) a new edge (\hat{v}_1, \tilde{v}) is added to E and the shape instance represented by node \hat{v}_1 becomes the template.

node in $\{\hat{v}_1, \hat{v}_2, \dots, \hat{v}_z\}$ has resulted in failure. If each candidate node resulted in failure, the subtree \tilde{T} rooted at node \tilde{v} is pruned from the MST and discarded.

3. Experiments

The proposed shape correspondence method was implemented in C++. The quadratic programming solver used to minimize the shape correspondence error was implemented using the OOQP C++ library developed by Gertz and Wright [9], the binary regions required to compute the Jaccard coefficient between two shape contours was implemented using the C++ OpenCV image processing library, and the minimum spanning tree was implemented using the C++ BOOST graph library and Kruskal’s algorithm. The CPU times reported in this section were acquired on Linux workstations running Intel Xeon 3.4GHz processor with 4GB of RAM. Additionally, the balance factor λ in Eq. (4), and the maximum number of iterations used by the quadratic programming solver for these experiments are set to 2 and 30 respectively.

The performance of the proposed method was evaluated using the shape correspondence benchmark introduced by Munsell *et al.* [17], and its performance was compared to four available global shape correspondence methods and one available pair-wise method. The four global methods are Thodberg’s implementation of the minimum description length method³ (T-MDL) [22][21], Ericsson and Karlsson’s implementation of the MDL method (E-MDL) [14], Ericsson and Karlsson’s implementation of the MDL method with curvature distance minimization (E-MDL+CUR) [14], and Ericsson and Karlsson’s implementation of the reparameterisation method by minimizing Euclidean distance (EUC) [14]. In these experiments the maximum number of iterations for these four global methods was set to 20, and we used their recommended settings, however for T-MDL

³Note that both T-MDL and E-MDL are based on a simplified description length cost function and are considered to be only approximate versions of MDL [8].

we do not allow the endpoints to move for open shape contours. The pair-wise shape correspondence method used in these experiments was Richardson and Wang’s implementation of a method that combines landmark sliding, insertion, and deletion (SDI) [19].

3.1. Shape Correspondence Benchmark

The performance of the proposed method was evaluated using the 2D shape correspondence benchmark in [17]. By introducing a ground-truth point distribution model [5], or PDM for short, the chosen benchmark allows for a more objective evaluation of shape correspondence performance than the previous compactness, generalization, and specificity measures of performance introduced by Davies *et al.* [7][20].

Four ground-truth PDMs were constructed that resemble the hand, corpus callosum (callosum for short), femur, and human face silhouette (face for short). These four ground-truth PDMs $\mathcal{N}(\bar{\mathbf{x}}^t, \mathbf{D}^t)$ are defined by a mean shape vector $\bar{\mathbf{x}}^t$ with dimension 256 and a covariance matrix \mathbf{D}^t with dimension 256×256 , i.e. each shape instance defines 128 landmarks. The mean shapes for the four ground truth shape structures are illustrated in Fig. 4. For each of the four ground-truth PDMs 800 synthetic shape instances were randomly generated using principal component analysis, where the values along each principal direction are multiplied by a random number drawn from a 1D Gaussian distribution $\mathcal{N}(0, 1)$. These 800 synthetic shape instances are randomly rotated, translated, scaled and then densely interpolated into continuous shape contours using the Catmull-Rom cubic spline and provided to the six test shape correspondence methods.



Figure 4. From left to right, ground-truth mean shapes that resemble the human hand, corpus callosum, and femur, and human face silhouette shape structures.

For each of the four shape structures, the test shape correspondence method identified 64 corresponded landmarks along the 800 synthetic continuous shape contours. The random scaling, rotation, and translation transformations introduced by the benchmark are then removed and four derived PDMs $\mathcal{N}(\bar{\mathbf{x}}, \mathbf{D})$ are constructed that define a mean shape vector $\bar{\mathbf{x}}$ with dimension 128, and covariance matrix \mathbf{D} with dimension 128×128 .

Shape correspondence performance is evaluated by measuring the bipartite Δ_b performance between the four ground-truth PDMs and the four derived PDMs [17], where Δ_b measures the difference in the probabilistic shape spaces defined by the ground-truth PDM and derived PDM. This is

accomplished by randomly generating 2000 synthetic shape contours using the process described above, however no random scaling, rotation, or translation transformations are introduced. The 2000 synthetic shape contours generated by the ground-truth PDM, and 2000 synthetic shape contours generated by the derived PDM are then used to construct a fully connected bipartite graph, where each node represents a synthetic shape instance, and the edge that connects two nodes is weighted by the Jaccard coefficient in Eq. (2). Using Goldberg and Kennedy’s cost scaling push relabeling algorithm [10] the bipartite Δ_b value is found, where Δ_b is limited to a value in $[0, 1]$. $\Delta_b = 0$ implies the two shape spaces defined by the derived and ground-truth PDMs are similar, and $\Delta_b = 1$ implies these two PDMs describe two very different shape spaces. To check the stability of the Δ_b measure, this random simulation process is repeated for 50 rounds. In Section 3.2, the reported Δ_b values for the 50 rounds of random simulation are averaged producing the final Δ_b performance values reported in Table 1.

3.2. Results

The resulting bipartite Δ_b performance for the six test shape correspondence methods using the Jaccard coefficient are shown in Table-1. Additionally, the 50 rounds of random simulation for the Δ_b measure are illustrated in Fig. 5. In Table-1 and Fig. 5 the performance of the proposed method is identified by “MST”. From Table-1 and Fig. 5, we see that the proposed method demonstrates the best performance for the ground-truth PDMs that resembles the callosum, and femur shape structures, has comparable performance to E-MDL for the ground-truth PDM that resembles the face, and has comparable performance to E-MDL and E-MDL+CUR for the ground-truth PDM that resembles the hand.

Table-2 contains the CPU time (in seconds) taken by the six test shape correspondence methods to correspond the 800 random continuous shape contours generated by each of the four ground-truth PDMs. From Table-2 we see that the proposed shape correspondence method is 62 times faster than E-MDL for femur shape structure, 46 times faster than E-MDL for the callosum shape structure, 36 times faster than E-MDL for the hand shape structure, and 42 times faster than E-MDL for the face shape structure. SDI is 4 times faster than the proposed method for each of the four shape structures, however from Table-1 we can see that SDI demonstrates the worst performance for the ground-truth PDMs that resembles the hand and femur shape structures, and demonstrates the second worst performance for the ground-truth PDM that resembles the callosum shape structure.

In general, the proposed method produces a shape correspondence that is better or comparable to the four available

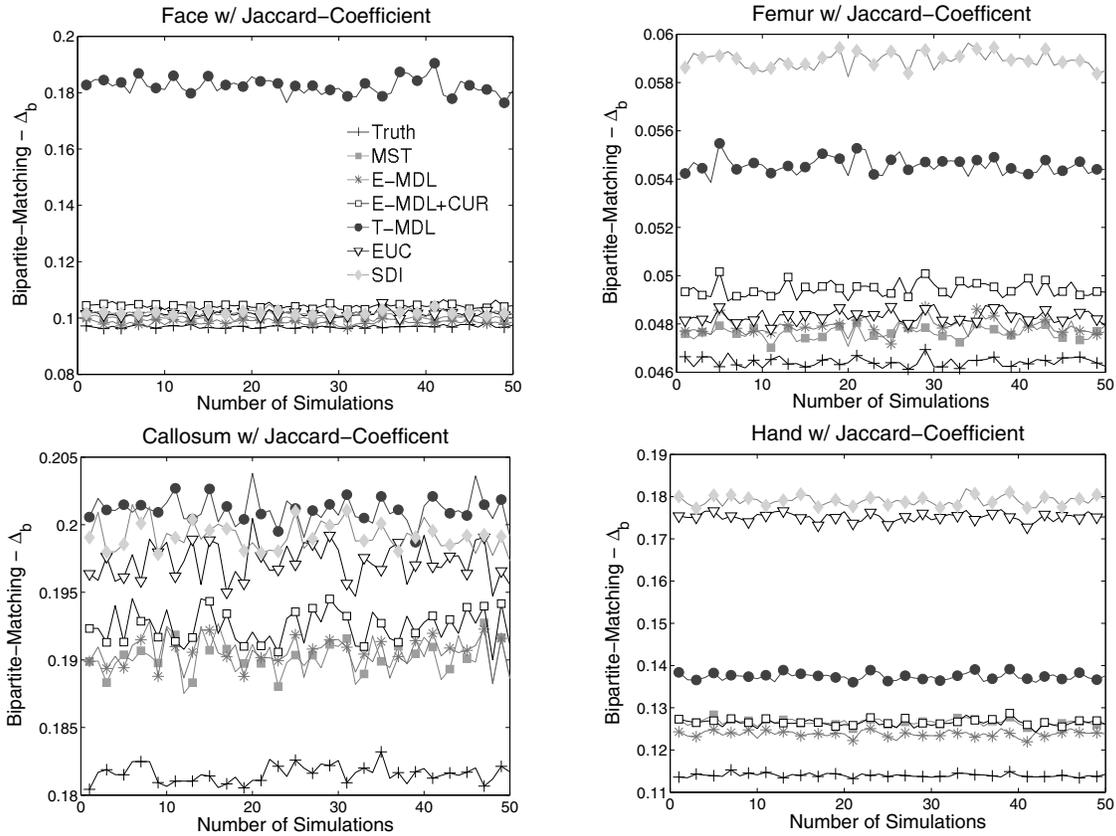


Figure 5. Δ_b values for the four ground-truth PDMs resulting from the six test shape correspondence methods using the Jaccard coefficient. The x -axis indicates the round of the random simulation. The curves with the “+” symbols are the values of Δ_b between each ground-truth PDM and itself.

global methods, and significantly better than the pair-wise method for the hand, callosum, and femur shape structures. Additionally, the proposed method is at least 30 times faster than the best global method E-MDL.

	Hand	Callosum	Femur	Face
MST	0.12656	0.19021	0.04762	0.10026
T-MDL	0.13733	0.20105	0.05455	0.18226
E-MDL	0.12381	0.19053	0.04788	0.09847
E-MDL+CUR	0.12646	0.19248	0.04943	0.10432
EUC	0.17520	0.19727	0.04830	0.10148
SDI	0.17897	0.19916	0.05893	0.10189

Table 1. Δ_b values resulting from the six test shape correspondence methods using the Jaccard coefficient. Each Δ_b value is the average value over the 50 random simulations.

4. Conclusion

In conclusion, this paper introduced a fast multiple shape correspondence method by pre-organizing shape instances. The population of shape instances are pre-organized by con-

	MST	T-MDL	E-MDL	E-MDL+CUR	EUC	SDI
Hand	2927	50784	107317	304504	29572	739
Callosum	2318	44732	107506	278832	28420	703
Femur	1757	59663	109875	261093	28538	740
Face	2417	50710	103822	259551	28286	745

Table 2. CPU time (in Seconds) required by the six test shape correspondence methods.

structing a minimum spanning tree where each node represents a shape instance and each edge connects two very similar shape instances. A root node is selected which represents the starting template shape instance then using the constructed MST and the selected root node we recursively traverse the MST in a depth-first fashion corresponding neighboring shape instances using a pair-wise method. We then evaluated the performance of the proposed shape correspondence method to four global methods and one pair-wise method. Quantitative evaluation showed the proposed method produces a shape correspondence that is better or comparable to the global methods and has better performance than the pair-wise method. Additionally, the pro-

posed method is at least 30 times faster than the global method that demonstrates the best performance.

Acknowledgements

This work was funded, in part, by AFOSR FA9550-07-1-0250. We would like to thank T. Thodberg for providing the Matlab version of T-MDL, and A. Ericsson for providing Matlab versions of E-MDL, E-MDL+CUR and EUC. We would also like to thank Jarrell Waggoner for all his help.

References

- [1] F. Bookstein. Principal warps: Thin-plate splines and the decomposition of deformations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(6):567–585, June 1989.
- [2] F. Bookstein. *Morphometric tools for landmark data*. Cambridge University Press, 1991.
- [3] F. Bookstein. Landmark methods for forms without landmarks: Morphometrics of group differences in outline shape. *Medical Image Analysis*, 1(3):225–243, 1997.
- [4] E. Catmull and R. Rom. A class of local interpolating splines. *Computer Aided Geometric Design*, pages 317–326, 1974.
- [5] T. Cootes, C. Taylor, D. Cooper, and J. Graham. Active shape models - their training and application. *Computer Vision and Image Understanding*, 61(1):38–59, Jan. 1995.
- [6] T. Cormen, C. Leiserson, and R. Rivest. *Introduction to Algorithms*. MIT Press, Cambridge, MA, 1990.
- [7] R. Davies. Learning shape: Optimal models for analysing natural variability. *Dissertation University of Manchester*, 2002.
- [8] R. Davies, C. Twining, T. Cootes, J. Waterton, and C. Taylor. A minimum description length approach to statistical shape modeling. *IEEE Transactions on Medical Imaging*, 21(5):525–537, May 2002.
- [9] E. M. Gertz and S. J. Wright. Object-oriented software for quadratic programming. *ACM Transactions on Mathematical Software*, 29:58–81, 2001.
- [10] A. Goldberg and R. Kennedy. An efficient cost scaling algorithm for the assignment problem. *Mathematic Programming*, 71:153–178, 1995.
- [11] J. Gower and G. Dijkstra. *Procrustes Problems*. Oxford University Press, 2004.
- [12] T. Heimann, I. Wolf, T. Williams, and H. P. Meinzer. 3d active shape models using gradient descent optimization of description length. In *Information Processing in Medical Imaging Conference*, 2005.
- [13] A. Hill and C. J. Taylor. Automatic landmark generation for point distribution models. In *British Machine Vision Conference*, volume 2, pages 429–438, 1994.
- [14] J. Karlsson and A. Ericsson. A ground truth correspondence measure for benchmarking. In *International Conference on Pattern Recognition*, volume 3, pages 568–573, 2006.
- [15] M. Leventon, E. Grimson, and O. Faugeras. Statistical shape influence in geodesic active contours. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 316–323, 2000.
- [16] D. Meier and E. Fisher. Parameter space warping: Shape-based correspondence between morphologically different objects. *IEEE Transactions on Medical Imaging*, 21:31–47, 2002.
- [17] B. C. Munsell, P. Dalal, and S. Wang. Evaluating shape correspondence for statistical shape analysis: A benchmark study. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(11):2023–2039, 2008.
- [18] M. Powell. A thin plate spline method for mapping curves into curves in two dimensions. In *Proc. Computational Techniques and Applications*, pages 43–57, 1995.
- [19] T. Richardson and S. Wang. Nonrigid shape correspondence using landmark sliding, insertion and deletion. In *International Conference on Medical Image Computing and Computer Assisted Intervention*, pages II–435–442, 2005.
- [20] M. Styner, K. Rajamani, L.-P. Nolte, G. Zsemlye, G. Szekely, C. Taylor, and R. Davies. Evaluation of 3D correspondence methods for model building. In *Information Processing in Medical Imaging Conference*, 2003.
- [21] H. Thodberg. Adding curvature to minimum description length shape models. In *British Machine Vision Conference*, volume 2, pages 251–260, 2003.
- [22] H. Thodberg. Minimum description length shape and appearance models. In *Information Processing in Medical Imaging Conference*, pages 51–62, 2003.
- [23] S. Wang, T. Kubota, and T. Richardson. Shape correspondence through landmark sliding. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages I–143–150, 2004.
- [24] J. Xie and P. Heng. Shape modeling using automatic landmarking. In *International Conference on Medical Image Computing and Computer Assisted Intervention*, pages II–709–716, 2005.