CHAPTER 3

Correspondence Establishment in Statistical Shape Modeling: Optimization and Evaluation

Song Wang*, Brent Munsell†, Theodor Richardson‡

*University of South Carolina, Department of Computer Science and Engineering, Columbia, SC 29208
†College of Charleston, Department of Computer Science, Charleston, SC 29401
‡South University, College of Business, Savannah, GA 31405

Contents

3.1 INTRODUCTION

Shape plays a critical role in medical imaging and image analysis. Many anatomic structures (organs and tissues) bear specific shapes and the accurate shape modeling can help segment these structures of interest from noisy medical images, which is a fundamental step for further medical image analysis and understanding. In addition, abnormal shape deformation may reflect pathological changes of a structure, which calls for accurate modeling of the normal shape variation of a specific structure. To model the shape and possible shape variation of a structure, statistical shape modeling [4,14] has been attracting considerable research interest in the past decades.

As in traditional statistics, where a statistical distribution is estimated to describe a set of data samples, in statistical shape modeling, a multi-dimensional statistical distribution

Statistical Shape and Deformation Analysis
DOI: 10.1016/B978-0-12-810493-4.00004-3

is estimated from a set of shape instances of the structure of interest to describe its shape and possible shape variation. Point distribution model (PDM) [4] is one of the most widely used statistical shape models, where a set of landmark points is identified from each shape instance and concatenating the coordinates of all the landmarks on each shape instance leads to a *shape vector*. Finally all the shape vectors are fitted to a multi-dimensional Gaussian distribution as the desired statistical shape model.

The main challenge in building a PDM is *shape correspondence*, which identifies the same number of landmarks on each shape instance such that landmarks across different shape instances are well corresponded [2,16,27,10]. Furthermore, the identified landmarks on each shape instance must represent the underlying shape very well. As a result, they must show certain level of spatial density and may not be located at anatomically meaningful points. In practice, shape of interest may be of different topology. In particular, the shape of a structure can be closed or open – in 2D it can be in the form of a closed curve or an open curve, while in 3D it can be in the form of a closed surface or an open surface. It is expected that a shape correspondence algorithm can handle both closed and open shapes.

In addition, we need to simultaneously correspond a set of shape instances instead of corresponding a pair of shape instances as in many shape-matching tasks. Typically, this is achieved by corresponding all the shape instances to a common template in a pairwise fashion. This way, shape correspondence performance and the accuracy of PDM construction are highly dependent on the selection of a good template [17,20]. Finally, the performance evaluation of shape correspondence is a challenging problem, since there is usually no unique ground-truth result for shape correspondence. People have used certain statistical properties of the resulting PDM to estimate the performance of shape correspondence, which is indirect and not fully objective [18,19].

In this chapter, we introduce a landmark-sliding based method for shape correspondence that can handle both closed and open shapes. Based on this, we study the groupwise shape correspondence that is not dependent on a common template. Finally, we introduce a more objective performance evaluation method for shape correspondence. For simplicity, we focus on discussing the 2D shape correspondence, with a brief discussion on its 3D extension at the end of this chapter. The models and methods introduced in this chapter are mainly drawn from a sequence of our past publications [26, 18,19,17,20].

The remainder of the chapter is organized as follows. Section 3.2 introduces the point distribution model (PDM) and the problem of shape correspondence. Section 3.3 introduces the landmark-sliding algorithm for pairwise shape correspondence. Section 3.4 discusses the groupwise shape correspondence by pre-organizing shape instances using a rooted tree. Section 3.5 discusses the performance evaluation of the shape correspondence. Section 3.6 reports some experiment results, followed by a brief discussion of the 3D extension and conclusions in Section 3.7.

3.2 PDM AND SHAPE CORRESPONDENCE

For simplicity, we focus on the 2D shape correspondence in this chapter. Given n sample shape instances (*shape contours* in the 2D case) S_i, $i = 1, 2, \ldots, n$ of the structure of interest, PDM [4] is usually constructed by three sequential steps: shape correspondence, shape normalization, and statistical modeling.

As discussed above, shape correspondence aims to identify corresponded landmarks from a set of continuous sample shape instances. More specifically, after shape correspondence we obtain n corresponded landmark sets V_i, $i = 1, 2, \ldots, n$, where $V_i = \{\mathbf{v}_{i1}, \mathbf{v}_{i2}, \ldots, \mathbf{v}_{im}\}$ are m landmarks identified from shape contour S_i and $\mathbf{v}_{ij} = (x_{ij}, y_{ij})$ is the jth landmark identified on S_i. Landmark correspondence means that $\mathbf{v}_{ij}$, $i = 1, 2, \ldots, n$, i.e., the jth landmark in each shape contour, are corresponded, for any $j = 1, 2, \ldots, m$.

In medical imaging, structural shape is usually assumed to be invariant to the transformations of any (uniform) scaling, rotation, and translations. In shape normalization, such transformations are removed among the given n shape contours by normalizing each of the n identified corresponded landmark sets V_i to $\hat{V}_i = \{\hat{\mathbf{v}}_{i1}, \hat{\mathbf{v}}_{i2}, \ldots, \hat{\mathbf{v}}_{im}\}$, $i = 1, 2, \ldots, n$. After the shape normalization, the absolute coordinates of the corresponded landmarks, i.e., $\hat{\mathbf{v}}_{ij} = (\hat{x}_{ij}, \hat{y}_{ij})$, $i = 1, 2, \ldots, n$ are directly comparable. Procrustes analysis [12] is the most widely used tool for shape normalization.

Finally, we construct the statistical shape model by fitting the normalized landmarks sets $\hat{V}_i = \{\hat{\mathbf{v}}_{i1}, \hat{\mathbf{v}}_{i2}, \ldots, \hat{\mathbf{v}}_{im}\}$, $i = 1, 2, \ldots, n$ to a multivariate Gaussian distribution. Specifically, we columnize m landmarks in $\hat{V}_i$ into a $2m$-dimensional vector $\hat{\mathbf{v}}_i = (\hat{x}_{i1}, \hat{y}_{i1}, \hat{x}_{i2}, \hat{y}_{i2}, \ldots, \hat{x}_{im}, \hat{y}_{im})^T$ and call it a *shape vector* of the shape instance $\hat{V}_i$. This way, the (landmark-based) mean shape $\bar{\mathbf{v}}$ and the covariance matrix $\mathbf{D}$ can be calculated by

$$\begin{aligned} \bar{\mathbf{v}} &= \frac{1}{n}\sum_{i=1}^{n}\hat{\mathbf{v}}_i \\ \mathbf{D} &= \frac{1}{n-1}\sum_{i=1}^{n}(\hat{\mathbf{v}}_i - \bar{\mathbf{v}})(\hat{\mathbf{v}}_i - \bar{\mathbf{v}})^T. \end{aligned} \tag{3.1}$$

The Gaussian distribution $\mathcal{N}(\bar{\mathbf{v}}, \mathbf{D})$ is the resulting PDM that models the possible shape-deformation space of the considered structure. Clearly, the accuracy of the PDM is largely dependent on the performance of shape correspondence, i.e., the accuracy in identifying the corresponded landmarks V_i, $i = 1, 2, \ldots, n$.

3.3 LANDMARK SLIDING FOR SHAPE CORRESPONDENCE

In this section, we focus on pairwise shape correspondence – corresponding one shape instance, say S_p, referred to as the target, to another shape instance, say S_q, referred to

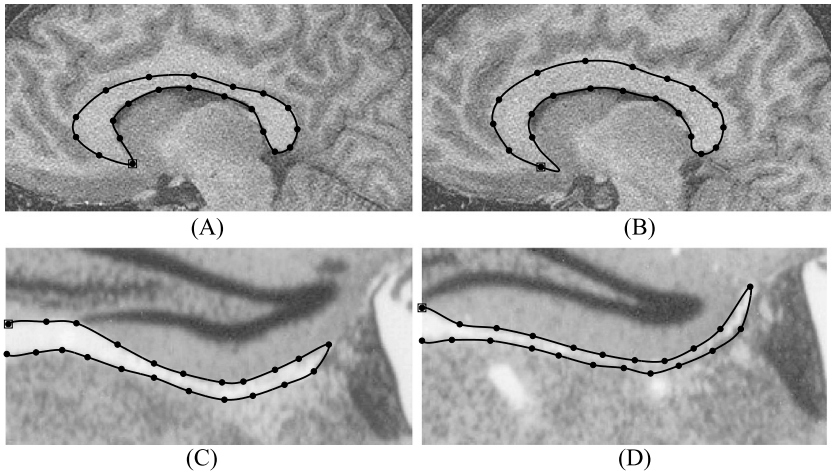

Figure 3.1 An illustrations of (A–B) closed-curve shape correspondence, and (C–D) open-curve shape correspondence. (A–B) are two Callosum shape instances and (C–D) are two stria shape instances in the human brain.

as the template. Extending pairwise correspondence to groupwise correspondence will be discussed in detail in the later Section 3.4.

For simplicity, we first consider the shape in the form of a closed curve, as illustrated in Fig. 3.1A–B, with known parametric representation. Furthermore, we assume that the landmarks in the template S_q are given and our task is to identify a set of corresponding landmarks in the target S_p. Later in the section, we will introduce the extension to open-shape correspondence, as illustrated in Fig. 3.1C–D.

Denote the parametric representation of the template S_q as $\mathbf{v}_q(t) = (x_q(t), y_q(t))^T$, $0 \le t \le L_q$, as shown in Fig. 3.1A, and the target S_p as $\mathbf{v}_p(s) = (x_p(s), y_p(s))^T$, $0 \le s \le L_p$, as shown in Fig. 3.1B, where L_q and L_p are the total curve lengths of the template and the target, respectively. With the assumption of closed shape, we have $\mathbf{v}_q(0) = \mathbf{v}_q(L_q)$ and $\mathbf{v}_p(0) = \mathbf{v}_p(L_p)$. In this parametrization, $t|L_q$ represents the traversed curve length from $\mathbf{v}_q(0)$ to $\mathbf{v}_q(t)$ and $s|L_p$ represents the traversed curve length from $\mathbf{v}_p(0)$ to $\mathbf{v}_p(s)$, where $a|b$ is the modulus operation.

Let $\mathbf{t} = \{t_0, t_1, \ldots, t_{m-1}\}$ be a set of parameters which generates m sequentially-sampled landmarks $V_q = \{\mathbf{v}_q(t_0), \mathbf{v}_q(t_1), \ldots, \mathbf{v}_q(t_{m-1})\}$. We assume that those landmarks represent the template shape contour S_q well. The task of shape correspondence is to find along the target shape contour S_p the corresponding m parameters $\mathbf{s} = \{s_0, s_1, \ldots, s_{m-1}\}$ such that the landmarks $V_p = \{\mathbf{v}_p(s_0), \mathbf{v}_p(s_1), \ldots, \mathbf{v}_p(s_{m-1})\}$ best match the landmarks V_q in the template. As a landmark representation, V_p must also well represent the underlying continuous contour of the target S_p.

Given the template landmarks, shape correspondence can be formulated as seeking the optimal parameters $\mathbf{s}$ to minimize the cost function

$$\phi(\mathbf{s}) = d(V_q, V_p) + \lambda R(\mathbf{s}),$$

where $d(V_q, V_p)$ measures the (landmark-based) shape difference and therefore reflects the global shape deformation between the template and target shape contours. $R(\mathbf{s})$ is the representation error when using the m landmarks V_p to represent the underlying target shape contour. We desire small representation error $R(\mathbf{s})$ such that landmark-based shape difference $d(V_q, V_p)$ accurately reflects the difference between the underlying continuous shape contours S_q and S_p. One additional constraint to be considered in this formulation is the preservation of the shape topology, i.e., the resulting landmarks $\mathbf{v}_p(s_0), \mathbf{v}_p(s_1), \ldots, \mathbf{v}_p(s_{m-1})$ should be sequentially located along the target shape contour as landmarks V_q located along the template shape contour.

3.3.1 Thin-Plate Splines to Measure Shape Difference

Following the notations in Section 3.2, we can write $\mathbf{v}_q(t_i)$ as $\mathbf{v}_{qi} = (x_{qi}, y_{qi})^T$ and $\mathbf{v}_p(s_i)$ as $\mathbf{v}_{pi} = (x_{pi}, y_{pi})^T$, $i = 0, 1, \ldots, m-1$. 2D thin-plate splines (TPS) has been widely used to measure the nonrigid deformation between two sets of the landmarks [1]. Specifically, 2D TPS defines a mapping $\mathbf{h} = (f, g)^T : \mathbb{R}^2 \to \mathbb{R}^2$ that maps the template landmarks V_q to the target landmarks V_p, i.e., $\mathbf{v}_{pi} = \mathbf{h}(\mathbf{v}_{qi}) = (f(\mathbf{v}_{qi}), g(\mathbf{v}_{qi}))$, $i = 0, 1, \ldots, m-1$ by minimizing the TPS bending energy [1]

$$\phi(\mathbf{h}) = \int\int_{-\infty}^{\infty} (L(f) + L(g))\,dxdy, \tag{3.2}$$

where $L(\cdot) = (\frac{\partial^2}{\partial x^2})^2 + 2(\frac{\partial^2}{\partial x \partial y})^2 + (\frac{\partial^2}{\partial y^2})^2$. The TPS bending energy can also be written as

$$\phi(\mathbf{h}) = \frac{1}{8\pi}(\mathbf{x}_p^T \mathbf{L} \mathbf{x}_p + \mathbf{y}_p^T \mathbf{L} \mathbf{y}_p),$$

where $\mathbf{L}$ is the $m \times m$ upper left submatrix of

$$\begin{pmatrix} \mathbf{K} & \mathbf{P} \\ \mathbf{P}^T & \mathbf{0} \end{pmatrix}^{-1}. \tag{3.3}$$

Here $\mathbf{K}$ is an $m \times m$ matrix with $k_{ij} = \|\mathbf{v}_{qi} - \mathbf{v}_{qj}\|^2 \log \|\mathbf{v}_{qi} - \mathbf{v}_{qj}\|$, $i, j = 0, 1, \ldots, m-1$. $\mathbf{P} = (\mathbf{1}, \mathbf{x}_q, \mathbf{y}_q)$ with $\mathbf{x}_q = (x_{q0}, x_{q1}, \ldots, x_{q,m-1})^T$, and $\mathbf{y}_q = (y_{q0}, y_{q1}, \ldots, y_{q,m-1})^T$. Similarly, $\mathbf{x}_p = (x_{p0}, x_{p1}, \ldots, x_{p,m-1})^T$, $\mathbf{y}_p = (y_{p0}, y_{p1}, \ldots, y_{p,m-1})^T$.

Note that $\mathbf{L}$ is a positive semidefinite matrix and the thin-plate bending energy is invariant to affine transforms, including scaling, rotation and translation, which is

desired for shape correspondence in medical-imaging applications. We can use this TPS bending energy to define the landmark-based shape difference, i.e.,

$$d(V_q, V_p) = \frac{1}{8\pi}(\mathbf{x}_p^T \mathbf{L}\mathbf{x}_p + \mathbf{y}_p^T \mathbf{L}\mathbf{y}_p). \tag{3.4}$$

3.3.2 Landmark-Based Shape Representation Error

Let l_{pi}, $i = 0, 1, \ldots m-1$ be the traversed curve length between two neighboring landmarks $\mathbf{v}_{pi}$ and $\mathbf{v}_{p,i+1|m}$ along the target shape contour. We have

$$l_{pi} = (s_{i+1|m} - s_i)|L_p.$$

Similarly, we have the corresponding curve length between neighboring landmarks in the template shape contour as

$$l_{qi} = (t_{i+1|m} - t_i)|L_q.$$

Not allowing two landmarks to coincide, we have

$$0 < l_{pi} < L_p, 0 < l_{qi} < L_q, \quad i = 0, 1, \ldots, m-1.$$

Given that $\mathbf{v}_{q0}, \mathbf{v}_{q1}, \ldots, \mathbf{v}_{q,m-1}$ are consecutively sampled landmarks along the closed template shape contour S_q, we have

$$\sum_{i=0}^{m-1} l_{qi} = L_q. \tag{3.5}$$

Given that the landmarks V_q well represent the underlying continuous contour S_q with sufficient spatial sampling density, we can require the landmarks in the target, i.e., V_p, to show a similar spatial distribution to well represent the continuous contour S_p, i.e.,

$$\frac{l_{pi}}{l_{p,i+1|m}} \approx \frac{l_{qi}}{l_{q,i+1|m}}, \quad i = 0, 1, \ldots, m-1. \tag{3.6}$$

Based on this, we can define the landmark representation error in the target shape contour as

$$R(\mathbf{s}) = \sum_{i=0}^{m-1} (l_{pi} l_{q,i+1|m} - l_{p,i+1|m} l_{qi})^2.$$

3.3.3 Topology Preservation and Landmark-Sliding Algorithm

To preserve the shape topology, we require the resulting landmarks $\mathbf{v}_{p0}, \mathbf{v}_{p1}, \ldots, \mathbf{v}_{p,m-1}$ to be consecutively located along the target shape contour as their corresponding landmarks in the template. Similar to (3.5), this can be reflected by the following constraint to the shape correspondence cost function,

$$\sum_{i=0}^{m-1} l_{pi} = L_p.$$

This way, the landmark-based shape-correspondence problem can be formulated as

$$\min_{\mathbf{s}} \left\{ (\mathbf{x}_p^T \mathbf{L} \mathbf{x}_p + \mathbf{y}_p^T \mathbf{L} \mathbf{y}_p) + \lambda \sum_{i=0}^{m-1} (l_{pi} l_{q,i+1|m} - l_{p,i+1|m} l_{qi})^2 \right\}, \tag{3.7}$$

subject to

$$\begin{aligned} & 0 < s_i < L_p \\ & l_{pi} = (s_{i+1|m} - s_i)|L_p, \quad i = 0, 1, \ldots, m-1 \\ & \sum_{i=0}^{m-1} l_{pi} = L_p, \end{aligned}$$

where $\lambda > 0$ is a regularization factor that balances the contribution from the shape difference and the shape representation error.

Finding the global optimal solution to this constrained optimization problem is difficult because of the nonlinearity inherent in the parametric representation of the shape contours. Instead, we develop an iterative algorithm to get a local optimal solution. Specifically, we initialize an $\mathbf{s}$ by sampling a sequence of m landmarks along the target shape contour S_p with the same spatial density and distribution as their corresponding landmarks along the template shape contour, i.e.,

$$\frac{l_{pi}}{L_p} = \frac{l_{qi}}{L_q}.$$

We then iteratively slide these initialized landmarks along the target shape contour to minimize the cost function (3.7) subject to the topology-preserving constraint.

Specifically, each iteration of the landmark sliding consists of two steps: (a) sliding all the landmarks along their respective tangent directions, and (b) projecting the updated landmarks back to the target shape contour. Let $\mathbf{s} = \{s_0, s_1, \ldots, s_{m-1}\}$ be the parameters for the currently estimated landmarks $V_p = \{\mathbf{v}_{p0}, \mathbf{v}_{p1}, \ldots, \mathbf{v}_{p,m-1}\}$. Each step of landmark sliding actually seeks an improved parameters $\mathbf{s}' = \{s'_0, s'_1, \ldots, s'_{m-1}\}$ for updated landmarks $V'_p = \{\mathbf{v}'_{p0}, \mathbf{v}'_{p1}, \ldots, \mathbf{v}'_{p,m-1}\}$.

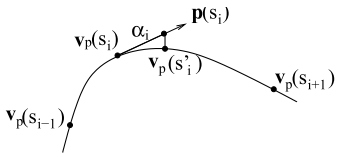

Figure 3.2 An illustration of the landmark sliding algorithm – each iteration consisting of a step of landmark sliding along the tangent direction, followed by a step of projection back to the shape contour.

Let $\mathbf{p}(s) = (p_x(s), p_y(s))$ be the (normalized) tangent direction of the target shape contour at $\mathbf{v}_p(s)$. As illustrated in Fig. 3.2, let α_i be the desired sliding distance of $\mathbf{v}_{pi}$ along the tangent direction. The step (a) moves the landmark $\mathbf{v}_{pi}$ to $\mathbf{v}_{pi} + \alpha_i \mathbf{p}(s_i)$ and the step (b) further projects it back onto the shape contour at

$$\mathbf{v}'_{pi} = \mathbf{v}_p(s_i + \alpha_i),$$

for all $i = 0, 1, \ldots, m-1$.

In step (a), we seek optimal sliding distance $\boldsymbol{\alpha} = (\alpha_0, \alpha_1, \ldots, \alpha_{m-1})^T$ to minimize the cost function (3.7) which can be rewritten as

$$\begin{aligned}&(\mathbf{x}_p + \mathbf{P}_x\boldsymbol{\alpha})^T\mathbf{L}(\mathbf{x}_p + \mathbf{P}_x\boldsymbol{\alpha}) + (\mathbf{y}_p + \mathbf{P}_y\boldsymbol{\alpha})^T\mathbf{L}(\mathbf{y}_p + \mathbf{P}_y\boldsymbol{\alpha})\\ &\quad + \lambda\sum_{i=0}^{m-1}\left\{(l_{pi} + \alpha_i)l_{q,i+1|m} - (l_{p,i+1|m} + \alpha_{i+1|m})l_{qi})^2\right\},\end{aligned} \tag{3.8}$$

where matrices $\mathbf{P}_x = \mathrm{diag}(p_x(s_0), p_x(s_1), \ldots, p_x(s_{m-1}))$ and $\mathbf{P}_y = \mathrm{diag}(p_y(s_0), p_y(s_1), \ldots, p_y(s_{m-1}))$. We can calculate the updated parameters $\mathbf{s}'$ by

$$s'_i = (s_i + \alpha_i)|L_p,$$

and the updated curve-segment length l'_{pi} by

$$l'_{pi} = (s'_{i+1} - s'_i)|L_p = (l_{pi} - \alpha_i + \alpha_{i+1})|L_p.$$

The shape-topology preservation constraint can be rewritten as

$$l_{pi} - \alpha_i + \alpha_{i+1} > 0, \quad i = 0, 1, \ldots, m-1, \tag{3.9}$$

i.e., no landmark can move beyond its neighbors during the sliding/projection.

It can be seen that minimizing the cost function (3.8) subject to constraints (3.9) is a simple quadratic programming problem, whose optima can be efficiently calculated. We can iteratively run steps (a) and (b) until convergence or a preset maximum number of iterations has reached.

3.3.4 Open-Shape Correspondence

Open shapes may occur when the structure of interest is occluded or cropped by image perimeter, as shown in Fig. 3.1C–D. The above landmark-sliding algorithm can be extended to handle the open-shape correspondence. Specifically, in the cost function (3.7), the shape-difference term will stay unchanged because the TPS bending energy is defined by only the landmarks. But the representation error term $R(\mathbf{s})$ and the additional constraints need to be updated to reflect open-curve topology.

Considering two open-shape instances S_p (target) and S_q (template), each of which is an open curve, the simplest case is that the endpoints are known to be corresponded between these two shape instances, as shown in Fig. 3.1C–D. This case can be addressed by making the following two changes to the above landmark-sliding algorithm: (a) the first landmark $\mathbf{v}_{p0}$ and the last landmark $\mathbf{v}_{p,m-1}$, i.e., the two endpoints of the curve, should be fixed in the landmark sliding. This can be enforced by additional linear constraints $\alpha_0 = 0$ and $\alpha_{m-1} = 0$; (b) In defining the representation error $R(\mathbf{s})$ and topology-preservation constraint, we remove the term corresponding to the curve segment between $\mathbf{v}_{p0}$ and $\mathbf{v}_{p,m-1}$.

A more complex case is that the endpoints do not correspond across the two open-shape instances. In this case, one shape instance may only be matched to a portion of the other shape instance. To adapt the landmark-sliding algorithm to handle this case, we need to allow the endpoints ($\mathbf{v}_{p0}$ and $\mathbf{v}_{p,m-1}$) to move along the shape curve, but limited to a preset maximum sliding distance in case all the landmarks on the target are aggregated together. More details on this case can be found from [21,22].

3.4 GROUPWISE SHAPE CORRESPONDENCE

In the last section, we introduce the landmark-sliding algorithm to correspond a pair of shape instances (either open or closed). In practice, we need to correspond a group of n shape instances for constructing a PDM, as discussed in Section 3.2. One simple strategy to achieve this goal [2] is to perform multiple rounds of the pairwise shape correspondence – we can select one shape instance as a template and then take each of the other $n-1$ shape instances as a target and correspond each target to the template independently using a pairwise correspondence algorithm as described in the last section. To further reduce the dependence on the initially selected template, we can compute the PDM, take the mean shape $\bar{\mathbf{v}}$ as the new template and correspond each of the n shape instance to this new template using the same pairwise correspondence algorithm. This process can be repeated in multiple iterations for a better shape correspondence and PDM. Since this is a direct application of the pairwise correspondence to handle a group of shape instances, we still call it a *pairwise* method in this chapter. While the pairwise correspondence is usually computational efficient, it may produce poor results when the shape instances to be corresponded show large variance [19].

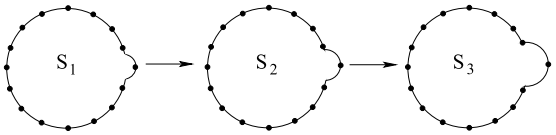

Figure 3.3 An illustration of the groupwise shape correspondence.

Another category of the methods for corresponding a group of n shape instances is to formulate a global cost function involving all the shape instances and then find landmarks on all the shape instances by minimizing the cost function. For example, Minimum Description Length (MDL) [10] method has been widely used for shape correspondence where the global cost function is defined as the total number of bits to represent all the shape instances and the constructed PDM. Another global shape correspondence method is Spherical Harmonics (SPHARM) [11], which is developed specifically for 3D closed shape by mapping each shape instance to a sphere and parameterizing the sphere. *Global* methods, such as MDL and SPHARM, are usually computationally intensive, especially in 3D case, because of the high complexity and nonlinearity in the optimized cost function.

In this section we introduce a *groupwise* correspondence method to address the limitations of global and pairwise methods. This method pre-organizes all the shape instances by constructing a minimum spanning tree (MST), where each node in the MST represents a shape instance and each edge in the MST connects two similar shape instances. A root node that represents the starting template shape instance is selected. All the shape instances that are the children of the root are taken as the targets and corresponded to the root using a pairwise method. Each of the child node is then taken as the template and corresponded to its own children using the same pairwise method. Recursively this will propagate the landmarks from the root to the leaf nodes to achieve the correspondence of all the shape instances. On one hand, this groupwise method is computationally efficient, because it performs a sequence of pairwise correspondences by following the MST. On the other hand, it can be more accurate and reliable than the pairwise method because in this groupwise method, the pairwise correspondence is only performed between similar shape instances. As shown in Fig. 3.3, following the MST, this groupwise method may choose to correspond S_2 to S_1 first, and then S_3 to S_2. Both shape pairs show good similarity, which may lead to higher correspondence accuracy than a pairwise method, e.g., corresponding both S_2 and S_3 to a common template S_1, since S_1 and S_3 show large difference.

3.4.1 Pre-Organizing Shape Instances using Minimum Spanning Tree

The basic goal of pre-organizing the shape instances is to identify shape pairs with high similarity. As in Section 3.2, let's consider a set of n shape instances S_i, $i = 1, 2, \ldots, n$ in

the form of continuous shape contours. We can independently correspond each pair of shape instances S_i and S_j by identifying m landmarks on each of them, indicated by V_i and V_j respectively. Using Procrustes analysis [12], we can normalize the landmarks V_i and V_j to $\hat{V}_i$ and $\hat{V}_j$, respectively. Applying the same normalization to the original shape contours S_i and S_j, we can obtain normalized shape contours $\hat{S}_i$ and $\hat{S}_j$, respectively. We define the dissimilarity between S_i and S_j as

$$w_{ij} = w(S_i, S_j) = \Delta(\hat{S}_i, \hat{S}_j) \times \Omega(V_i, V_j) \tag{3.10}$$

for each pair of (i, j), $i \neq j$, $i, j = 1, 2, \ldots, n$.

In Eq. (3.10), the first term $\Delta(\hat{S}_i, \hat{S}_j)$ is the Jaccard coefficient

$$\Delta(\hat{S}_i, \hat{S}_j) = 1 - \frac{|\mathcal{R}(\hat{S}_i) \cap \mathcal{R}(\hat{S}_j)|}{|\mathcal{R}(\hat{S}_i) \cup \mathcal{R}(\hat{S}_j)|}, \tag{3.11}$$

where $\mathcal{R}(S)$ is the region enclosed by contour S, and $|\mathcal{R}|$ computes the area of the region $\mathcal{R}$. For open shapes, the first and last landmarks are connected to enclose the region. The Jaccard coefficient takes a value between $[0, 1]$ with zero indicating that the two shape instances are exactly coincident with each other.

The second term $\Omega(V_i, V_j)$ measures the landmark-based nonrigid shape difference between S_i and S_j. We can use the TPS bending energy as defined in Section 3.3. More specifically, we define

$$\Omega(V_i, V_j) = \frac{d(V_i, V_j) + d(V_j, V_i)}{2},$$

where $d(V_i, V_j)$ and $d(V_j, V_i)$ are the TPS bending energies [1] as defined in Eq. (3.4).

Given n shape instances, we can then build a fully connected graph with n nodes, with each node representing a shape instance. Between each pair of nodes, say S_i and S_j, we connect them by an edge, with edge weight w_{ij} as defined in Eq. (3.10). By applying an MST algorithm, such as Prim's or Kruskal's algorithms [5], we can obtain an MST which connects all the n nodes using $n - 1$ edges with minimum total edge weight, as shown in Fig. 3.4. In the MST, only pair of nodes with relative small edge weight, i.e., only pair of shape instances with high similarity, are directly connected by an edge.

3.4.2 Root-Node Selection and Groupwise Shape Correspondence

In the MST, we select a root node as the initial template to start the correspondence (to the child nodes) and then recursively propagate the pairwise correspondence layer-by-layer until getting to the leaf nodes. Clearly, any shape-correspondence errors in the early recursions (upper layers) may be propagated to the later recursions (lower layers). Therefore, we need to select a root node such that in the MST, the edges with smaller depths, i.e., closer to the root, have smaller edge weights.

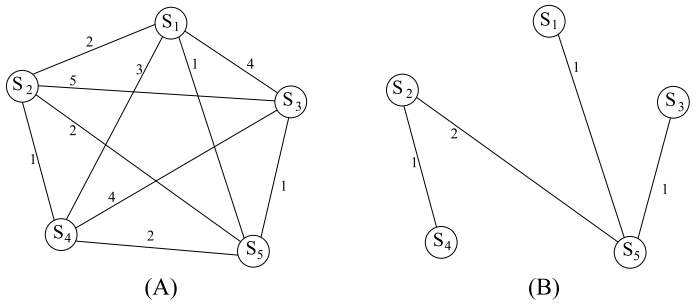

Figure 3.4 An illustration of a fully-connected graph (A) and its MST (B). Number on each edge is its weight.

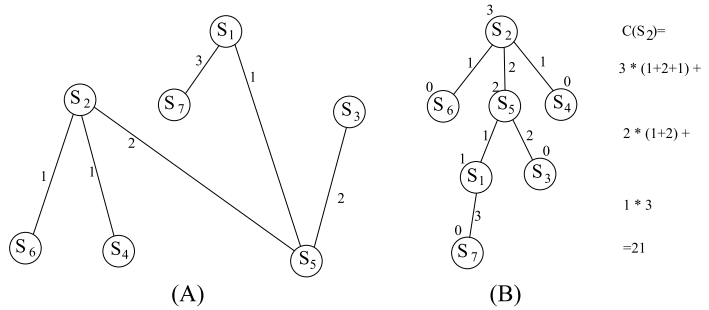

Figure 3.5 An illustration of the rooted MST and the root node selection. (A) An MST derived from the graph. (B) Selecting S_2 as root leads to a root cost of 21. Number on each edge is its weight and number on each node is its height.

To achieve this goal, we define the following root cost function $C(S_k)$ for each node (shape instance) S_k, $k = 1, 2, \ldots, n$. By selecting S_k as the root, we can draw the rooted MST, denoted as $MST(S_k)$, layer by layer as shown in Fig. 3.5. Denote all the edges in $MST(S_k)$ as (S_i, S_j) such that S_i is the parent node of S_j. For each node S_i in $MST(S_k)$ we further define its height $h(S_i)$ to be the number of edges along the longest path from this node S_i to a leaf node. As shown in Fig. 3.5, we define the root cost function

$$C(S_k) = \sum_{(S_i, S_j) \in MST(S_k)} w_{ij} \cdot h(S_i).$$

Then we select the shape instance (node)

$$\arg\min_{S_k} C(S_k)$$

as the root note of the MST for groupwise shape correspondence.

Based on the rooted MST, we can recursively propagate the landmarks from the root to each leaf and achieve the groupwise shape correspondence. Specifically, using the landmark-sliding algorithm described in Section 3.3, the groupwise shape correspondence algorithm consists of the following steps:

1. Sample landmarks along the root shape instance as the template.
2. Take each child node of the root in the MST as the target and correspond each target to the template using the landmark-sliding algorithm.
3. All the target shape instances in Step 2, with their identified landmarks, are then treated as the new templates. For each new template, find its children in the rooted MST as the new targets and perform landmark-sliding algorithm for shape correspondence, i.e., identifying corresponded landmarks on the new targets.
4. Repeat Step 3 to propagate the landmarks layer by layer in the rooted MST until all the leaf nodes are processed.

3.5 PERFORMANCE EVALUATION OF SHAPE CORRESPONDENCE

Objective and quantitative evaluation of shape correspondence is a challenging problem, because of the lack of the ground truth and the non-uniqueness in selecting the corresponded landmarks. Davies et al. [23] suggested the use of three general metrics – compactness, specificity, and generality – for evaluating shape-correspondence performance in terms of the PDM construction. These three metrics describe the properties of the PDM constructed from the identified landmarks – it follows the steps in Section 3.2 to construct a PDM $\mathcal{N}(\bar{\mathbf{v}}, \mathbf{D})$, and then defines the three metrics as the function of the number of considered principal eigenvectors of $\mathbf{D}$. However, it has been shown that the same shape space may show different values of compactness, generality, and specificity when represented by different sets of corresponded landmarks [19].

Using synthesized shape contours from a given PDM can well address these issues and achieve more objective performance evaluation. Specifically, we start from a given ground-truth PDM, from which we can randomly generate a set of synthetic shape contours. The shape-correspondence algorithm to be evaluated is applied to identify corresponded landmarks from these shape contours and leads to a PDM by following the steps in Section 3.2. The better the resulting PDM reflects the ground-truth PDM, the better the shape-correspondence performance.

As shown in Fig. 3.6, the performance evaluation consists of the following five components: (C1) specifying a PDM $\mathcal{N}(\bar{\mathbf{v}}^t, \mathbf{D}^t)$ as the ground truth, (C2) using this PDM to randomly generate a set of shape contours $S_1, S_2, \ldots, S_n$, (C3) running the test shape-correspondence algorithm on these shape contours to identify corresponded landmarks, (C4) deriving a PDM $\mathcal{N}(\bar{\mathbf{v}}, \mathbf{D})$ from the identified landmarks using Eq. (3.1), and (C5) comparing the derived PDM $\mathcal{N}(\bar{\mathbf{v}}, \mathbf{D})$ to the ground-truth PDM $\mathcal{N}(\bar{\mathbf{v}}^t, \mathbf{D}^t)$

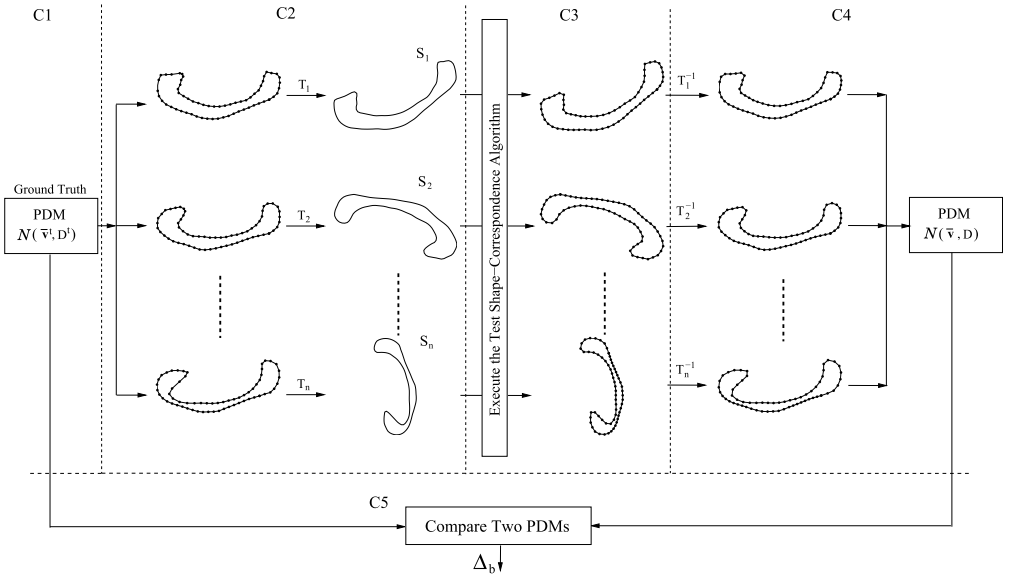

Figure 3.6 An illustration of the proposed shape-correspondence evaluation method.

and using their difference to measure the performance of the test shape-correspondence algorithm.

In essence, this approach evaluates the shape-correspondence algorithm's capability to recover the underlying statistical shape model from a set of sampled shape instances. This reflects the role of the shape correspondence in statistical shape modeling. In these five components, (C3) and (C4) are for shape correspondence and PDM construction, which have been discussed in detail in previous sections. The task of Component (C1) is to specify a mean shape vector $\bar{\mathbf{v}}^t$ and a covariance matrix $\mathbf{D}^t$. Ideally, they can take any values only if $\mathbf{D}^t$ is positive definite. In practice, we can pick them to resemble some real structures. In this section, we focus on developing algorithms for Components (C2) and (C5).

3.5.1 Generating Shape Instances

Given the ground-truth PDM $\mathcal{N}(\bar{\mathbf{v}}^t, \mathbf{D}^t)$ with k landmarks (k might be different from m, the number of landmarks identified by shape correspondence in Component (C3)), we can randomly generate as many sample shape vectors $\hat{\mathbf{v}}_i^t$, $i = 1, 2, \ldots, n$ as possible. More specifically, with $\hat{\mathbf{p}}_j^t$ and $\hat{\lambda}_j^t$, $j = 1, 2, \ldots, 2k$ being the eigenvectors and eigenvalues of $\mathbf{D}^t$, we can generate shape instances in the form of

$$\hat{\mathbf{v}}^t = \bar{\mathbf{v}}^t + \sum_{j=1}^{2k} \hat{b}_j^t \hat{\mathbf{p}}_j^t, \tag{3.12}$$

where $\hat{b}_j^t$ is independently and randomly sampled from the 1D Gaussian distribution $\mathcal{N}(0, \hat{\lambda}_j^t)$, $j = 1, 2, \ldots, 2k$.

Each shape vector $\hat{\mathbf{v}}_i^t$, $i = 1, 2, \ldots, n$ in fact defines k landmarks $\{\hat{\mathbf{v}}_{i1}^t, \hat{\mathbf{v}}_{i2}^t, \ldots, \hat{\mathbf{v}}_{ik}^t\}$. By assuming that these k landmarks are sequentially sampled from a continuous shape contour, we can estimate this continuous contour $\hat{S}_i$ by landmark interpolation. For constructing a closed shape contour, we interpolate the portion between the last landmark $\hat{\mathbf{v}}_{ik}^t$ and the first landmark $\hat{\mathbf{v}}_{i1}^t$. For constructing an open shape contour, we do not interpolate the portion between $\hat{\mathbf{v}}_{ik}^t$ and $\hat{\mathbf{v}}_{i1}^t$. We use the Catmull–Rom cubic spline [3] for interpolating these landmarks into a contour.

For each synthetic shape contour $\hat{S}_i$, we then apply a random affine transformation T_i, consisting of a random rotation, a random (uniform) scaling and a random translation to simulate the real cases in practice, to obtain shape contour S_i. We record the affine transformations T_i, $i = 1, 2, \ldots, n$ and then pass $S_1, S_2, \ldots, S_n$ to the test shape-correspondence algorithm. Note that the recorded affine transformations T_i, $i = 1, 2, \ldots, n$ are not passed to the test shape-correspondence algorithm (Component (C3)), because in real applications, these transformations are not priorly known and it is expected that the shape correspondence algorithm can handle them. If the test shape-correspondence algorithm introduces further transformations, such as Procrustes analysis, in Component (C3), we record and undo these transformations before outputting the shape-correspondence result. This ensures the corresponded landmarks identified by the test shape-correspondence algorithm are placed directly back onto the input shape contours $S_1, S_2, \ldots, S_n$. Then in Component (C4), we directly apply the inverse transform T_i^{-1}, $i = 1, 2, \ldots, n$, to the landmarks identified on S_i. This guarantees the correct removal of the random affine transformation T_i before PDM construction in Component (C4).

3.5.2 Difference Between Two PDMs

The PDM $\mathcal{N}(\bar{\mathbf{v}}, \mathbf{D})$ derived in Component (C4) and the ground-truth PDM $\mathcal{N}(\bar{\mathbf{v}}^t, \mathbf{D}^t)$ each represents a shape space. The closer these two shape spaces, the better the shape correspondence that is used to construct PDM $\mathcal{N}(\bar{\mathbf{v}}, \mathbf{D})$. However, directly computing a distance metric between these two PDMs, e.g., a Kullback–Leibler distance [15], is not applicable in this task, because the landmarks in $\bar{\mathbf{v}}$ and $\bar{\mathbf{v}}^t$ are not corresponded. Actually, $\bar{\mathbf{v}}$ and $\bar{\mathbf{v}}^t$ may be of different dimensions because they may consist of different number of landmarks, i.e., $\bar{\mathbf{v}} \in \mathbb{R}^{2m}$, $\bar{\mathbf{v}}^t \in \mathbb{R}^{2k}$ and $m \neq k$, where m and k are the number of landmarks along each shape contour in these two PDMs.

To compare the underlying shape spaces, we compare the two PDMs in the continuous shape space instead of using the sampled landmarks. Specifically, we use a random-simulation strategy: randomly generating a large set of N shape vectors from each PDM using Eq. (3.12), interpolating these landmarks defined by these shape vectors into continuous shape contours using Catmull–Rom cubic spline, and then

measuring the similarity between these two sets of shape contours. We denote the N continuous shape contours generated from PDM $\mathcal{N}(\bar{\mathbf{v}}, \mathbf{D})$ to be $\hat{S}_1^c, \hat{S}_2^c, \ldots, \hat{S}_N^c$ and the N continuous shape contours generated from the ground-truth PDM $\mathcal{N}(\bar{\mathbf{v}}^t, \mathbf{D}^t)$ to be $\hat{S}_1^t, \hat{S}_2^t, \ldots, \hat{S}_N^t$. When N is sufficiently large, the difference between these two sets of continuous shape contours can well reflect the difference of the shape spaces underlying these two PDMs.

Given a pair of continuous shape contours $\hat{S}_i^c$ and $\hat{S}_j^t$, generated from two PDMs respectively, we can define their difference by Jaccard coefficient $\Delta(\hat{S}_i^c, \hat{S}_j^t)$ using Eq. (3.11), because the possible affine transformation between these two shape contours have been removed as described in Section 3.5.1. Based on this, we can estimate the difference between two shape-contour sets $\{S_i^c\}_{i=1}^N$ and $\{S_j^t\}_{j=1}^N$ using the bipartite-matching algorithm. We run bipartite matching between these two shape-contour sets so that the total matching cost, which is defined as the total difference between the matched shape contours, is minimal. Then we define a difference measure between these two PDMs as

$$\Delta_b \triangleq \frac{\sum_{i=1}^N \Delta(\hat{S}_i^c, \hat{S}_{b(i)}^t)}{N}, \tag{3.13}$$

where $\hat{S}_i^c$ and $\hat{S}_{b(i)}^t$ are the matched pair of shape contours in the bipartite matching. In this difference measure, we introduce a normalization over N so that Δ_b takes values in the range of $[0, 1]$. Using the bipartite-matching algorithm, the measure Δ_b not only assesses whether the two shape spaces (defined by the two PDMs) contain similar shape contours, but also examines whether a shape contour shows a similar probability density in these two shape spaces. The smaller the difference Δ_b, the better the shape-correspondence performance.

3.6 EXPERIMENTS

In this section, we evaluate the performance of the shape-correspondence algorithm based on landmark sliding, as described in Section 3.3 and Section 3.4, using the measure Δ_b as described in Section 3.5. The landmark-sliding algorithm was implemented in C++ and the balance factor λ in Eq. (3.7), and the maximum number of iterations used by the quadratic programming solver for these experiments, are set to 2 and 30 respectively. For comparison, we also evaluate the performance of several other shape correspondence algorithms. CPU time reported in this section for each evaluated shape correspondence algorithm is obtained from Linux workstations running Intel Xeon 3.4 GHz processor with 4 GB of RAM.

Specifically, we evaluate six algorithms in the experiments. *SDI*: the pairwise shape correspondence algorithm where all the shape instances are corresponded to a common

Figure 3.7 From left to right, ground-truth mean shapes that resemble the human hand, corpus callosum, and femur, and human face silhouette shape structures.

template by pairwise landmark-sliding algorithm. Initial template is a randomly selected shape instance and in later iterations the template is updated to be the mean shape computed from the previous iteration as described at the beginning of Section 3.4. *MST*: the minimum spanning tree (MST) based groupwise correspondence algorithm as described in Section 3.4. *T-MDL*: Thodberg's implementation of MDL with approximations [25,24]. *E-MDL*: Ericsson and Karlsson's implementation of the MDL [13]. *E-MDL+CUR*: Ericsson and Karlsson's implementation of the MDL method with curvature distance minimization [13]. *EUC*: Ericsson and Karlsson's implementation of the parameterization method by minimizing Euclidean distance [13].

As described in Section 3.5, we need to specify a ground-truth PDM to compute Δ_b for evaluating the shape correspondence performance. Four ground-truth PDMs were specified that resemble the hand, corpus callosum (callosum for short), femur, and human face silhouette (face for short) for the performance evaluation. These four ground-truth PDMs $\mathcal{N}(\bar{\mathbf{v}}^t, \mathbf{D}^t)$ are defined by a mean shape vector $\bar{\mathbf{v}}^t$ with dimension 256 and a covariance matrix $\mathbf{D}^t$ with dimension 256×256, i.e. each shape instance consists of $k = 128$ landmarks. The mean shapes for the four ground-truth PDMs are shown in Fig. 3.7. For each of the four ground-truth PDMs $n = 800$ synthetic shape contours were generated by following the steps described in Section 3.5. These 800 shape contours are used to test the performance of the six shape correspondence algorithms. All the test shape correspondence algorithms are set to identify $m = 64$ landmarks along each shape contour. $N = 2000$ synthetic shape contours are generated from both the PDM $\mathcal{N}(\bar{\mathbf{v}}^t, \mathbf{D}^t)$ and the PDM $\mathcal{N}(\bar{\mathbf{v}}, \mathbf{D})$ for estimating the difference Δ_b by following the steps described in Section 3.5. To check the stability of the Δ_b measure, this random simulation process of generating shape instances is repeated for 50 rounds.

For each shape-correspondence algorithm and each ground-truth PDM, the Δ_b values from the 50 rounds of random simulations are shown in Fig. 3.8. Their average Δ_b values over 50 rounds of simulations are shown in Table 3.1. We can see that the proposed groupwise method, i.e., MST, 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.

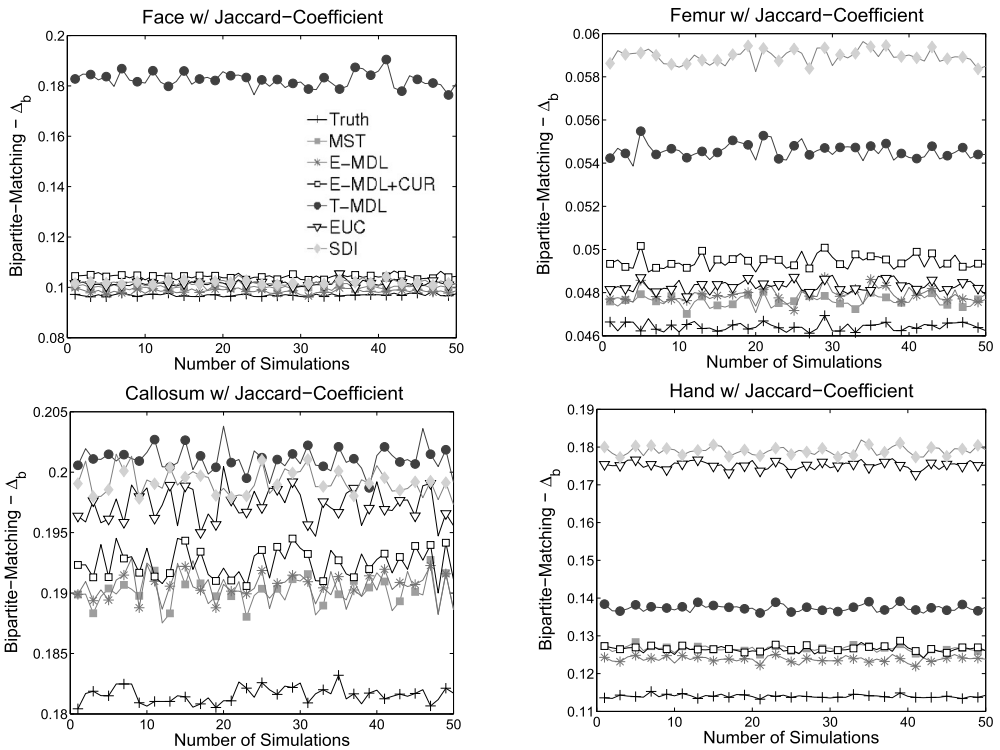

Figure 3.8 Δ_b values for the four ground-truth PDMs resulting from the six test shape correspondence methods using the Jaccard coefficient. The *x*-axis indicates the round of the random simulation. The curves with the "+" symbols are the values of Δ_b between each ground-truth PDM and itself.

Table 3.1 Δ_b values resulting from the six test shape correspondence algorithms. Each Δ_b value is the average value over the 50 random simulations

	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 3.2 reports the CPU time (in seconds) taken by the six test shape correspondence algorithms to correspond the $n = 800$ continuous shape contours generated by each of the four ground-truth PDMs. We can see that the proposed groupwise shape correspondence algorithm, e.g., MST, is much faster than the four global shape correspondence algorithms, i.e., T-MDL, E-MDL, E-MDL+CUR, and EUC. The pairwise

Table 3.2 CPU time (in seconds) taken by the six test shape correspondence methods

	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

SDI algorithm is about 4 times faster than the proposed MST algorithm. However, from Table 3.1 we can see that SDI demonstrates the worst Δ_b performance for the ground-truth PDMs that resemble the hand and femur shape structures, and demonstrates the second worst performance for the ground-truth PDM that resembles the callosum shape structure.

3.7 CONCLUSIONS AND 3D SHAPE CORRESPONDENCE

Shape correspondence, i.e., identifying a set of landmark points across a population of the shape instances, is one of the most challenging steps in constructing statistical shape models. In this chapter, we focused on discussing about the 2D shape correspondence and its performance evaluation. Specifically, we introduced a landmark-sliding algorithm that can move the landmarks along the shape contours to get better correspondence. We further introduced a groupwise correspondence algorithm that pre-organizes all the shape instances into a rooted minimum spanning tree and the landmarks are propagated from the root shape instance to the leaf nodes layer-by-layer using pairwise correspondence. Finally, we introduced more objective method to evaluate shape correspondence by comparing the shape spaces underlying the ground-truth PDM and the PDM constructed by the shape correspondence. Comparison experiments against several other well known shape correspondence algorithms were reported using this performance evaluation method.

With 3D medical imaging, such as MRI and CT, we can get 3D shape instances of many organs and structures. This leads to 3D statistical shape modeling and 3D shape correspondence. Just like 2D shape, 3D shape can be closed or open, as illustrated in Fig. 3.9. The increased geometric complexity and data size in 3D further increase the computational burden of shape correspondence. For example, global methods, such as MDL and SPHARM, have been used for 3D shape correspondence, but taking extensive CPU time. The landmark-sliding algorithm introduced in Section 3.3 can be easily extended to handle 3D shape correspondence to achieve a good balance between accuracy and running time. Please refer to [7,6,9] for more details on 3D landmark-sliding algorithms. The basic ideas underlying the groupwise correspondence algorithm based on MST in Section 3.4 and the performance evaluation method in Section 3.5 are also applicable to 3D shape [8].

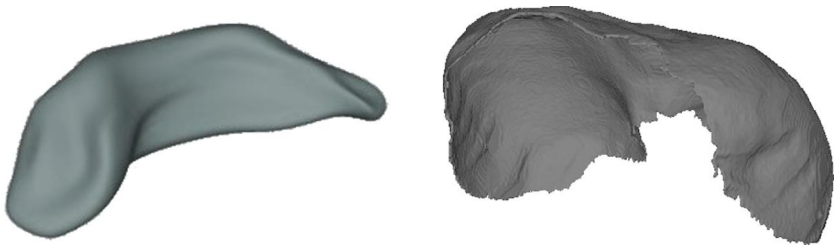

Figure 3.9 3D shape of the hippocampus (left) and diaphragm (right) structures. Note that hippocampus has a closed shape while diaphragm has an open shape.

REFERENCES

[1] F. Bookstein, Principal warps: thin-plate splines and the decomposition of deformations, IEEE Trans. Pattern Anal. Mach. Intell. 11 (6) (1989) 567–585.

[2] F. Bookstein, Landmark methods for forms without landmarks: morphometrics of group differences in outline shape, Med. Image Anal. 1 (3) (1997) 225–243.

[3] E. Catmull, R. Rom, A class of local interpolating splines, Comput. Aided Geom. Des. (1974) 317–326.

[4] T. Cootes, et al., Active shape models – their training and application, Comput. Vis. Image Underst. 61 (1) (1995) 38–59.

[5] T. Cormen, et al., Introduction to Algorithms, MIT Press, Cambridge, MA, 1990.

[6] P. Dalal, et al., 3D open-surface shape correspondence for statistical shape modeling: identifying topologically consistent landmarks, in: International Conference on Computer Vision, 2009.

[7] P. Dalal, et al., A fast 3D correspondence method for statistical shape modelling, in: IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2007.

[8] P. Dalal, et al., Multiple cortical surface correspondence using pairwise shape similarity, in: International Conference on Medical Image Computing and Computer Assisted Intervention, vol. 1, 2010, pp. 349–356.

[9] P. Dalal, S. Wang, Landmark Sliding for 3D Shape Correspondence, IGI Global, 2012.

[10] R. Davies, et al., A minimum description length approach to statistical shape modeling, IEEE Trans. Med. Imaging 21 (5) (2002) 525–537.

[11] G. Gerig, et al., Shape analysis of brain ventricles using SPHARM, in: IEEE Workshop on Mathematical Methods in Biomedical Image Analysis, 2001, pp. 171–178.

[12] J. Gower, G. Dijksterhuis, Procrustes Problems, Oxford University Press, 2004.

[13] J. Karlsson, A. Ericsson, A ground truth correspondence measure for benchmarking, in: Int. Conf. Pattern Recognit., vol. 3, 2006, pp. 568–573.

[14] D. Kendall, et al., Shape and Shape Theory, John Wiley & Sons, Ltd., 1999.

[15] S. Kullback, R.A. Leibler, On information and sufficiency, Ann. Math. Stat. 22 (1) (1951) 79–86.

[16] C. Lorenz, N. Krahnstover, Generation of point-based 3D statistical shape models for anatomical objects, Comput. Vis. Image Underst. 77 (2000) 175–191.

[17] B. Munsell, et al., Fast multiple shape correspondence by pre-organizing shape instances, in: IEEE Conference on Computer Vision and Pattern Recognition, 2009, pp. 840–847.

[18] B.C. Munsell, et al., A new benchmark for shape correspondence evaluation, in: International Conference on Medical Image Computing and Computer Assisted Intervention, 2007, pp. 507–514.

[19] B.C. Munsell, et al., Evaluating shape correspondence for statistical shape analysis: a benchmark study, IEEE Trans. Pattern Anal. Mach. Intell. 30 (11) (2008) 2023–2039.

[20] B.C. Munsell, et al., Pre-organizing shape instances for landmark-based shape correspondence, Int. J. Comput. Vis. 97 (2012) 210–228.
[21] T. Richardson, S. Wang, Nonrigid shape correspondence using landmark sliding, insertion and deletion, in: International Conference on Medical Image Computing and Computer Assisted Intervention, Part II, 2005, pp. 435–442.
[22] T. Richardson, S. Wang, Open-curve shape correspondence without endpoint correspondence, in: International Conference on Medical Image Computing and Computer Assisted Intervention, Part I, 2006, pp. 17–24.
[23] M. Styner, et al., Evaluation of 3D correspondence methods for model building, in: Information Processing in Medical Imaging Conference, 2003.
[24] H. Thodberg, Adding curvature to minimum description length shape models, in: British Machine Vision Conference, vol. 2, 2003, pp. 251–260.
[25] H. Thodberg, Minimum description length shape and appearance models, in: Information Processing in Medical Imaging Conference, 2003, pp. 51–62.
[26] S. Wang, et al., Shape correspondence through landmark sliding, Part I, in: IEEE Conference on Computer Vision and Pattern Recognition, 2004, pp. 143–150.
[27] J. Xie, P. Heng, Shape modeling using automatic landmarking, Part II, in: International Conference on Medical Image Computing and Computer Assisted Intervention, 2005, pp. 709–716.