Covariance Matrix of A Shape Population: A Tale on Spline Setting

Kang Li
Mechanical, Materials and Aerospace
Engineering Department
Illinois Institute of Technology, Chicago, IL 60616
Email: kli@hawk.iit.edu
Xiaoping Qian[1]
Department of Mechanical Engineering
University of Wisconsin, Madison WI 53706
Email: qian@engr.wisc.edu

Abstract

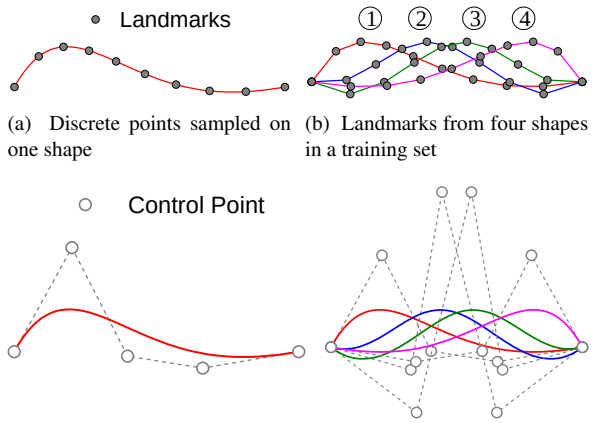
Computing the covariance matrix of a population of shapes is essential for establishing shape correspondence, identifying shape variation across the population, and building statistical shape models. The covariance matrix is usually computed from a discrete set of points (a.k.a. landmarks) sampled on each shape. The distribution and density of the sampled points thus greatly influence the covariance matrix and its spectral decomposition. To understand and overcome this dependency on point sampling, in this paper, we develop accurate and efficient methods for directly computing continuous formulations of the covariance matrix. We adopt B-splines both as a shape representation and as a form of reparameterization. We apply B-splines into two continuous formulations for computing the covariance matrix of shapes. We develop both analytical and efficient numerical methods for computing such matrices. In both formulations, the covariance matrix from the corresponding numerical approximation converges to the matrix from the continuous formulations when the number of sampled points in each shape becomes sufficiently large. Their applications in optimizing shape correspondence by minimizing the description length of a set of shapes are presented.

Keywords: B-splines, covariance matrix, shape statistics

1. Introduction

The growing use of 3D shape acquisition tools and rapid advancement of shape modeling techniques have led to increasing interest in shape modeling of a population of objects. Computing the covariance matrix of a population of shapes is essential for identifying shape variation across the population and building statistical shape models (SSMs). This statistical shape model provides a compact characterization of the shape variability pattern in a set of shapes (training set). Computing the covariance matrix for modeling a population of objects has witnessed growing applications including image segmentation [1][2], facial recognition[3], computer animation[4], medical diagnosis [5][6], patient-specific modeling [7][8], mass customization [9] and biological growth modeling, etc. Shape variation is usually extracted through a statistical technique, *Principal Component Analysis* (PCA). This process is performed by eigenvalue decomposition of the covariance matrix. The eigenvectors of the decomposition characterize the geometric variation pattern, and the corresponding eigenvalues represent the amount of such shape variation. The covariance matrix is usually computed from a discrete set of points (a.k.a. *landmarks*) sampled on each shape, leading to the classical *Point Distribution Model* (PDM) [10]. Figure 1(a) displays discrete points sampled on a shape and such a point set from a collection of shapes displayed in Fig. 1(b) are then used to compute the covariance matrix.

In this paper, we propose the computing of the covariance matrix on a spline based continuous representation of shapes. More specifically, it involves the use of B-splines as a parametric shape representation (Figure 1(c) and (d)). It also involves the use of B-splines



(a) Discrete points sampled on one shape

(b) Landmarks from four shapes in a training set

(c) Continuous B-spline representation of a shape

(d) B-spline representation of four shapes

Figure 1: Discrete and continuous representations of shapes for computing the covariance matrix.

to represent the reparameterization of parametric curves and surfaces. Note that B-spline based shape representation and B-spline based reparameterizations are independent and they can have different numbers of control points, knot vectors and degrees. The use of B-splines for shape representation and for reparameterization is motivated by the following considerations. B-splines provide a compact and flexible parametric shape representation. B-splines are also capable of representing various reparameterization functions due to their local modification property. More importantly, the dual use of B-splines makes it straightforward to compute directly from the formulations of the covariance matrix of continuous shapes. In these formulations of the covariance matrix of a shape population, the input for each matrix entry is two continuous shapes, rather than discrete points on the two shapes. We refer to these formulations as *continuous formulations*. Although the continuous formulations [11, 12, 13, 14] have been developed for some time by extending the discrete form from finite number of points to infinite number of points, the forms themselves are deemed hard to compute. In [15], the entries in the continuous form of the covariance matrix were approximated by finite points weighted by the area on the mean shape, and it was found that such an approximated continuous form is effective in ensuring faithful sampling of shapes and preventing the sampled point from moving away from "difficult " areas on the shapes.

In this paper, we apply our B-spline representations into two continuous formulations of the covariance matrix [13, 14]. We indicate that, with B-splines, these continuous forms can be either computed in an analytical form or approximated through numerical integration. Two common numerical integration scheme-based approaches, the mid-point and Gauss quadratures, are given for computing the covariance matrix. We also demonstrate that, with these two continuous formulations, one is amendable for analytical computing (without discretization), but is parameterization-dependent and the other is parameterization-independent and more stable in computing the shape correspondence for building SSM. We apply the B-spline based continuous formulations of the covariance matrix in the optimization of the correspondence across a shape population. This is achieved by reparameterizing each shape to minimize the description length (DL) of the shape population.

The contribution of this paper is that, with B-splines, these continuous forms of covariance matrix are readily computable. Both the closed-form and the quadrature based numerical procedure are efficient and accurate in the sense it would take a large number of sampled points with the usual discrete landmark points based formulation to converge to the same covariance matrix. *When data points are parameterized with the chord length method in B-spline fitting, the resulting covariance matrix does not depend on the data sampling scheme.*

The remainder of this paper is organized as follows. In Section 2, we review the usual discrete formulation for computing the covariance matrix and the continuous formulations from the literature. In Section 3, we apply the B-spline-based shape representation and reparameterization into two continuous formulations and derive closed-form and efficient quadrature methods for computing the covariance matrix. Section 4 presents closed-form and quadrature methods for computing the covariance matrix from reparameterized B-spline curves/surfaces. In Section 5, we briefly present the optimization formulation for minimizing the description length of a shape population with the computed covariance matrix. In Section 6, we present our numerical results, for which we compare the results from the two formulations and then apply the continuous formulation of the covariance matrix in optimizing shape correspondence for building SSMs. This paper is concluded in Section 7.

2. Review: discrete and continuous formulations of the covariance matrix

In this section, we briefly summarize the covariance matrix formulations that have been proposed in literature. They can be categorized into discrete and continuous formulations.

2.1. Discrete formulation

We first review currently the most commonly used discrete formulation for computing the covariance matrix. For more in-depth information about the current use of the covariance matrix in statistical shape modeling, refer to [2, 13].

For a training set of n_S shapes $\{\mathcal{S}_i\}$ $(i = 1, 2, ..., n_S)$, the discrete formulation assumes n_P landmarks on each shape so that i-th shape $\mathcal{S}_i$ is approximated by the *shape vector*

$$\mathbf{X}_i \doteq \left[\mathbf{x}_i^{(1)}, \mathbf{x}_i^{(2)}, ..., \mathbf{x}_i^{(n_P)}\right]^T, \tag{1}$$

where every landmark is sampled and lies strictly on the shape, namely: $\mathbf{x}_i^{(j)} = \left[x_i^{(j)}, y_i^{(j)}, z_i^{(j)}\right] \in \mathcal{S}_i;\ \forall j \in 1, 2, ..., n_P$.

The *discrete covariance matrix* is defined by

$$\mathbf{D} \doteq \frac{1}{n_S - 1} \sum_{i=1}^{n_S} \left(\mathbf{X}_i - \bar{\mathbf{X}}\right)\left(\mathbf{X}_i - \bar{\mathbf{X}}\right)^T, \tag{2}$$

where $n_S - 1$ is the divider for unbiased sample covariance and the *discrete mean shape* is

$$\bar{\mathbf{X}} \doteq \frac{1}{n_S} \sum_{i=1}^{n_S} \mathbf{X}_i. \tag{3}$$

Concatenating the *mean-removed* shape vectors forms the shape *data matrix* defined as follows

$$\mathcal{X} = [\mathbf{X}_1 - \bar{\mathbf{X}},\ \mathbf{X}_2 - \bar{\mathbf{X}},\ ...,\ \mathbf{X}_{n_S} - \bar{\mathbf{X}}], \tag{4}$$

which is of size $3n_P \times n_S$; the covariance matrix (2) could then be compactly expressed by

$$\mathbf{D} = \frac{1}{n_S - 1} \mathcal{X}\mathcal{X}^T. \tag{5}$$

The *Principal Component Analysis* (PCA) [16] is frequently used to decompose the covariance matrix into the principal modes of shape variability via eigen-decomposition

$$\mathbf{D}\mathbf{v}_m = \lambda_m \mathbf{v}_m, \tag{6}$$

where $\{\lambda_m\}(m = 1, 2, ...n_S - 1)$ are all the non-zero eigenvalues such that $\lambda_1 \geqslant \lambda_2 ... \geqslant \lambda_{n_S - 1}$, and $\{\mathbf{v}_m\}$ the associated eigenvectors or eigenmodes. The mean shape $\bar{\mathbf{X}}$ and modes $\{\mathbf{v}_m\}$ along with eigenvalues $\{\lambda_m\}$ constitute the statistical model, which can provide a more compact representation of the shape variability of the shape population than the original training set.

An alternative for the covariance matrix definition is the following,

$$\widetilde{\mathbf{D}} \doteq \frac{1}{n_S - 1} \sum_{i=1}^{n_S} \left(\mathbf{X}_i - \bar{\mathbf{X}}\right)^T \left(\mathbf{X}_i - \bar{\mathbf{X}}\right) \tag{7a}$$

$$= \frac{1}{n_S - 1} \mathcal{X}^T \mathcal{X}, \tag{7b}$$

where the entry-wise definition for $\widetilde{D}$ is as follows

$$\widetilde{D}_{i_1 i_2} \doteq \frac{1}{n_S - 1} \left(\mathbf{X}_{i_1} - \bar{\mathbf{X}}\right)^T \left(\mathbf{X}_{i_2} - \bar{\mathbf{X}}\right), \tag{8}$$

and $\widetilde{\mathbf{D}} = \{\widetilde{D}_{i_1 i_2}\}$ $(i_1, i_2 = 1, 2, ..., n_S)$. Eigen-decomposition of the $\widetilde{\mathbf{D}}$ reveals its eigenvalues $\widetilde{\lambda}$ and eigenvectors $\widetilde{\mathbf{v}}$ with those of the original covariance matrix $\mathbf{D}$ as

$$\lambda_m = \widetilde{\lambda}_m, \tag{9a}$$

$$\mathbf{v}_m = \mathcal{X}\,\widetilde{\mathbf{v}}_m,\ \widetilde{\mathbf{v}}_m = \mathcal{X}^T \mathbf{v}_m. \tag{9b}$$

Thus the $n_S \times n_S$ covariance matrix $\widetilde{\mathbf{D}}$ and its original form $\mathbf{D}$ of size $3n_P \times 3n_P$ have the exact same non-zero eigenvalues from (9a) and eigenvectors that can be mutually converted by (9b).

This formulation featuring (2) or (7a) computes the covariance matrix by directly placing a finite set of landmarks sampled on each shape, which is also the core of the classical Point Distribution Model [10] in statistical shape modeling. It is referred to as the "*discrete formulation*" in this paper.

2.2. Continuous formulations

Suppose the training set shapes all possess the parameterization defined over the common parameter domain $\mathcal{U}$. The i-th shape $\mathcal{S}_i$ is parameterized with function $\mathbf{S}_i$ that maps a parameter point $\mathbf{u} \in \mathcal{U}$ to a point $\mathbf{x} \in \mathbb{R}^3$ on the shape in the physical domain.

2.2.1. Continuous formulation I

The entry of the $n_S \times n_S$ covariance matrix $\mathbf{C}^I$ for formulation I is defined as

$$C^I_{i_1 i_2} \doteq \frac{1}{n_S - 1} \int_{\mathcal{U}} \left[\mathbf{S}_{i_1}(\mathbf{u}) - \bar{\mathbf{S}}(\mathbf{u})\right]^T \left[\mathbf{S}_{i_2}(\mathbf{u}) - \bar{\mathbf{S}}(\mathbf{u})\right] \mathrm{d}\mathbf{u}, \tag{10}$$

where the mean shape is

$$\bar{\mathbf{S}}(\mathbf{u}) \doteq \frac{1}{n_S} \sum_{i=1}^{n_S} \mathbf{S}_i(\mathbf{u}). \tag{11}$$

$\mathbf{C}^I = \{C^I_{i_1 i_2}\}(i_1, i_2 = 1, ..., n_S)$ is characterized by the inner product between two mean-removed continuous shapes $\left[\mathbf{S}_{i_1}(\mathbf{u}) - \bar{\mathbf{S}}(\mathbf{u})\right]$ and $\left[\mathbf{S}_{i_2}(\mathbf{u}) - \bar{\mathbf{S}}(\mathbf{u})\right]$. This continuous formulation I for the curve case first appeared in [11] and is adopted in [14] for cardinal spline curves.

2.2.2. Continuous formulation II

This continuous formulation II considers directly the training set shape geometry and is parameterization independent; it is defined as follows

$$C^{II}_{i_1 i_2} \doteq \frac{\int_{\mathcal{U}} \left[\mathbf{S}_{i_1}(\mathbf{u}) - \bar{\mathbf{S}}(\mathbf{u})\right]^T \left[\mathbf{S}_{i_2}(\mathbf{u}) - \bar{\mathbf{S}}(\mathbf{u})\right] |\mathbf{J}(\mathbf{u})| \, \mathrm{d}\mathbf{u}}{(n_S - 1) \int_{\mathcal{U}} |\mathbf{J}(\mathbf{u})| \, \mathrm{d}\mathbf{u}}, \tag{12}$$

where the additional Jacobian term $|\mathbf{J}(\mathbf{u})|$ is the determinant of the Jacobian of a point on the mean shape. The term "$|\mathbf{J}(\mathbf{u})| \, \mathrm{d}\mathbf{u}$" corresponds to the "area measure for integration $\mathrm{d}\mu(\mathbf{u})$" in [15], and it is equivalent to the "length/area" element $\mathrm{d}A(\mathbf{x})$ in [13]. The difference in the formulation (12) is the normalization with the denominator, which makes it easier to examine the convergence properties and the influence of parameterization on the continuous formulation, as demonstrated in Section 6.

For curves, the Jacobian term is

$$|\mathbf{J}(u)| = \left| \frac{\mathrm{d}\, \bar{\mathbf{S}}(\mathbf{u})}{\mathrm{d}\, u} \right|, \tag{13}$$

and for surfaces, the Jacobian term is

$$|\mathbf{J}(u, v)| = \left| \frac{\partial\, \bar{\mathbf{S}}(u, v)}{\partial u} \times \frac{\partial\, \bar{\mathbf{S}}(u, v)}{\partial v} \right|. \tag{14}$$

3. Covariance matrix of spline curves and surfaces

For the above continuous formulations of the covariance matrix, (10) and (12), we demonstrate that they can be computed efficiently and accurately with Bézier/B-spline based shape representation, either in closed-form or with quadrature methods.

3.1. Continuous formulation I with analytical integral

With training set shapes in the Bézier form, the integrand in continuous formulation I (10) is the multiplication of two Bernstein polynomials, and thus the integration has analytical form. Shapes in the form of B-splines, i.e. a collection of Béziers, also have analytical integration for (10).

3.1.1. Bézier curves

We start by considering a collection of shapes that are represented by degree p Bézier curves. The i-th Bézier curve is defined by

$$\mathbf{S}_i(u) = \sum_{j=0}^{p} B^p_j(u) \mathbf{P}^{(i)}_j, \qquad u \in [0, 1] \tag{15}$$

where $B^p_j(u)$ is the degree p Bernstein basis for the j-th control point $\mathbf{P}^{(i)}_j (j = 0, 1, ..., p)$. The mean shape is

$$\bar{\mathbf{S}}(u) = \frac{1}{n_S} \sum_{i=1}^{n_S} \mathbf{S}_i(u) = \sum_{j=0}^{p} B^p_j(u) \bar{\mathbf{P}}_j, \tag{16}$$

which is still a degree p Bézier curve with control points

$$\bar{\mathbf{P}}_j = \frac{1}{n_S} \sum_{i=1}^{n_S} \mathbf{P}^{(i)}_j. \tag{17}$$

Similarly, the mean-removed i-th shape $\left[\mathbf{S}_i(u) - \bar{\mathbf{S}}(u)\right]$ is also a Bézier curve with control points

$$\widehat{\mathbf{P}}^{(i)}_j = \mathbf{P}^{(i)}_j - \bar{\mathbf{P}}_j.$$

The covariance matrix as formulated in (10) becomes

$$\begin{aligned} C^I_{i_1 i_2} =& \frac{1}{n_S - 1} \int_0^1 \left[\mathbf{S}_{i_1}(u) - \bar{\mathbf{S}}(u)\right]^T \left[\mathbf{S}_{i_2}(u) - \bar{\mathbf{S}}(u)\right] \mathrm{d}u \\ =& \frac{1}{n_S - 1} \int_0^1 \sum_{j_1=0}^{p} \sum_{j_2=0}^{p} B^p_{j_1}(u) B^p_{j_2}(u) \, \widehat{\mathbf{P}}^{(i_1)T}_{j_1} \widehat{\mathbf{P}}^{(i_2)}_{j_2} \mathrm{d}u \\ =& \frac{1}{n_S - 1} \int_0^1 \sum_{j=0}^{2p} B^{2p}_j(u) Q^{(i_1, i_2)}_j \, \mathrm{d}u, \end{aligned}$$

where

$$Q^{(i_1, i_2)}_j = \sum_{l=\max(0, j-p)}^{\min(j, p)} \frac{\binom{p}{l}\binom{p}{j-l}}{\binom{2p}{j}} \widehat{\mathbf{P}}^{(i_1)T}_l \widehat{\mathbf{P}}^{(i_2)}_{j-l}. \tag{18}$$

In the above equation, we utilize the fact that the product of two Bernstein polynomials of degree p and q is a higher order Berstein polynomial of degree $(p + q)$ as proved in [17], i.e.,

$$B^p_i(u) B^q_j(u) = \frac{\binom{p}{i}\binom{q}{j}}{\binom{p+q}{i+j}} B_{i+j, p+q}(u).$$

In this case multiplication of two degree$p$ Bézier curves becomes a degree $2p$ Bézier curve with new control points $Q^{(i_1, i_2)}_j$. By further considering the integral property of the Bernstein polynomial below [17]

$$\int_0^1 \sum_{j=0}^{p} B^p_j(u) Q_j \, \mathrm{d}u = \frac{\sum_{j=0}^{p} Q_j}{p + 1}, \tag{19}$$

the covariance matrix expression reduces to

$$C^I_{i_1 i_2} = \sum_{j=0}^{2p} \sum_{l=\max(0, j-p)}^{\min(j, p)} \frac{\binom{p}{l}\binom{p}{j-l} \widehat{\mathbf{P}}^{(i_1)T}_l \widehat{\mathbf{P}}^{(i_2)}_{j-l}}{\binom{2p}{j}(n_S - 1)(2p + 1)}. \tag{20}$$

Consequently, the covariance matrix for a Bézier represented shape population in the continuous formulation I can be obtained analytically without even resorting to sampling landmarks on the shapes.

3.1.2. Bézier surfaces

Here we assume that the shapes are represented in Bézier surfaces of degree p and q along the u- and v-direction, respectively. The i-th shape is a Bézier surface defined by

$$\mathbf{S}_i(\mathbf{u}) = \sum_{j=0}^{p}\sum_{k=0}^{q} B_j^p(u)B_k^q(v)\mathbf{P}_{j,k}^{(i)}, \qquad u \times v \in [0,1]^2, \tag{21}$$

where $B_j^p(u)$ and $B_k^q(v)$ are the Bernstein basis functions of degree p and q and $\mathbf{P}_{j,k}^{(i)}$ ($j = 0, 1, ..., p;\ k = 0, 1, ..., q$) is the control points.

Through the derivation similar to the Bézier curve case, the covariance matrix entry reduces to

$$C_{i_1 i_2}^{I} = \sum_{j=0}^{2p}\sum_{k=0}^{2q}\sum_{l=\max(0,j-p)}^{\min(j,p)}\sum_{m=\max(0,k-q)}^{\min(k,q)} \frac{\binom{p}{l}\binom{p}{j-l}\binom{q}{m}\binom{q}{k-m}\widehat{\mathbf{P}}_{l,m}^{(i_1)T}\widehat{\mathbf{P}}_{j-l,k-m}^{(i_2)}}{\binom{2p}{j}\binom{2q}{k}(n_S-1)(2p+1)(2q+1)}. \tag{22}$$

where $\widehat{\mathbf{P}}_{j,k}^{(i)} = \mathbf{P}_{j,k}^{(i)} - \bar{\mathbf{P}}_{j,k}$ is the Bézier control points for the mean-removed shape $\left[\mathbf{S}_i(u,v) - \bar{\mathbf{S}}(u,v)\right]$ for the i-th shape and $\bar{\mathbf{P}}_{j,k} = \frac{1}{n_S}\sum_{i=1}^{n_S}\mathbf{P}_{j,k}^{(i)}$.

Because a B-spline curve (surface) is simply a piecewise collection of Bézier curves (surfaces), the piecewise summation of (20) and (22) gives the analytical integrals for shapes represented in B-splines.

3.2. Continuous formulation II with analytical integrand

Due to the Jacobian involved in continuous formulation II (12), even for shapes that are parameterized by Béziers or B-splines, the analytical integration is difficult to obtain in general. However, the analytical form of the integrand can still be obtained, which would facilitate the quadrature based numerical integration.

3.2.1. Bézier curves

If all shapes are represented in Bézier curves of degree p with the definition presented in (15). The Jacobian is degree $(p-1)$ Bézier defined by

$$\mathbf{J}(u) = \sum_{k=0}^{p-1} B_k^{p-1}(u)\widetilde{P}_k, \tag{23}$$

where the new control points are

$$\widetilde{P}_k = p(\bar{P}_{k+1} - \bar{P}_k). \tag{24}$$

Therefore the covariance matrix in this context reduces to

$$C_{i_1 i_2}^{II} = \frac{\int_0^1 \sum_{j=0}^{2p}\sum_{k=0}^{p-1} B_j^{2p}(u)B_k^{p-1}(u)Q_j^{(i_1,i_2)}\widetilde{P}_k \,\mathrm{d}u}{(n_S-1)\int_0^1 \sum_{k=0}^{p-1} B_k^{p-1}(u)\widetilde{P}_k \,\mathrm{d}u}, \tag{25}$$

where $Q_j^{(i_1,i_2)}$ is defined earlier in (18).

3.2.2. Bézier surfaces and B-spline curves/surfaces

Similarly the analytical form of $\mathbf{J}(u)$ for Bézier surface, B-spline curves and B-spline surfaces can be obtained. Thus the integrand in the second continuous formulation of the covariance matrix can be obtained exactly for B-spline curves and surfaces.

3.3. Approximation of continuous formulation I and II

The numerical quadrature based approximation of the continuous formulations leads to what will be referred to as the "*approximated continuous forms*" of the covariance matrix, not to be confused with the discrete formulation in (8). Such approximated continuous forms would reveal the link between the continuous form and the usual discrete form. We examine two integration schemes below: the mid-point rule and the Gaussian quadrature.

3.3.1. Continuous formulation I

Integration by mid-point rule. The integration in the curve case by mid-point rule is achieved by discretizing the parameter domain $\mathcal{U} = [0,1]$ into n_P parameter intervals $[\widetilde{u}_j, \widetilde{u}_{j+1}](j = 1, ..., n_P)$ of equal length, where $\widetilde{u}_1 = 0, \widetilde{u}_{n_P+1} = 1$. Then, the approximate covariance matrix is

$$\widetilde{C}_{i_1 i_2}^{I,Mid} = \frac{1}{n_S-1}\sum_{j=1}^{n_P}\left[\mathbf{S}_{i_1}(\xi_j) - \bar{\mathbf{S}}(\xi_j)\right]^T\left[\mathbf{S}_{i_2}(\xi_j) - \bar{\mathbf{S}}(\xi_j)\right]\Delta u_j,$$

where the mid-point parameter is $\xi_j = (\widetilde{u}_j + \widetilde{u}_{j+1})/2$. Due to equal intervals and recalling that the point $\mathbf{S}(\xi_j)$ on the i-th shape can be regarded as the j-th landmark $\mathbf{x}_i^{(j)}$ as in (1), the above could be written as

$$\begin{aligned}\widetilde{C}_{i_1 i_2}^{I,Mid} &= \frac{1}{n_S-1}\sum_{j=1}^{n_P}\left[\mathbf{x}_{i_1}^{(j)} - \bar{\mathbf{x}}\right]^T\left[\mathbf{x}_{i_2}^{(j)} - \bar{\mathbf{x}}\right]\frac{1}{n_P} \\ &= \frac{1}{(n_S-1)n_P}\left(\mathbf{X}_{i_1} - \bar{\mathbf{X}}\right)^T\left(\mathbf{X}_{i_2} - \bar{\mathbf{X}}\right).\end{aligned} \tag{26}$$

This demonstrates that the discrete formulation (2) or (7a) is equivalent to the approximated continuous formulation I (26)

$$\widetilde{C}^{I,Mid}_{i_1 i_2} = \frac{1}{n_P}\widetilde{D}_{i_1 i_2}. \tag{27}$$

which holds for the surface case as well by means of similar derivation.

Integration by Gaussian quadrature. In addition to the mid-point rule to evaluate the integral in (10), an additional common form of numerical integration is applied through Gaussian quadrature. Because the integrand in (10) is just degree $2p$ polynomials for curves, a Gaussian quadrature with at least $n_G \geq p+1$ quadrature points is expected to give the exact answer. Similar conclusions can be drawn for surfaces. In our implementation, each knot span of B-spline curves corresponds to [-1,1] interval for Gaussian quadratures.

3.3.2. Continuous formulation II

The above numerical integration approaches can be similarly used to obtain the approximate forms for continuous formulation II.

Integration by mid-point rule. The covariance matrix of the continuous formulation II in (12) for the curve case can be approximated by

$$\widetilde{C}^{II,Mid}_{i_1 i_2} = \frac{\sum_{j=1}^{n_P}\left[\mathbf{S}_{i_1}(\xi_j) - \bar{\mathbf{S}}(\xi_j)\right]^T\left[\mathbf{S}_{i_2}(\xi_j) - \bar{\mathbf{S}}(\xi_j)\right]\Delta L(\xi_j)}{(n_S-1)\sum_{j=1}^{n_P}\Delta L(\xi_j)}, \tag{28}$$

where the weight over each segment is the discretized arc length at the evaluated point $\bar{\mathbf{S}}(\xi_j)$ on the mean shape defined by

$$\Delta L(\xi_j) = \begin{cases} \frac{|\bar{\mathbf{S}}(\xi_{j+1}) - \bar{\mathbf{S}}(\xi_j)|}{2} & j = 1 \\ \frac{|\bar{\mathbf{S}}(\xi_j) - \bar{\mathbf{S}}(\xi_{j-1})| + |\bar{\mathbf{S}}(\xi_{j+1}) - \bar{\mathbf{S}}(\xi_j)|}{2} & 1 < j < n_P \\ \frac{|\bar{\mathbf{S}}(\xi_j) - \bar{\mathbf{S}}(\xi_{j-1})|}{2} & j = n_P \end{cases} \tag{29}$$

The approximate covariance matrix of continuous formulation II in the surface case is defined by

$$\widetilde{C}^{II,Mid}_{i_1 i_2} = \frac{\sum_{j=1}^{n_{Pu}}\sum_{k=1}^{n_{Pv}}\widehat{\mathbf{S}}_{i_1}(\xi_j,\eta_k)\widehat{\mathbf{S}}_{i_1}(\xi_j,\eta_k)\Delta A(\xi_j,\eta_k)}{(n_S-1)\sum_{j=1}^{n_{Pu}}\sum_{k=1}^{n_{Pv}}\Delta A(\xi_j,\eta_k)}, \tag{30}$$

where the (j,k)-th point on the mean-removed shape is

$$\widehat{\mathbf{S}}_i(\xi_j,\eta_k) = \mathbf{S}_i(\xi_j,\eta_k) - \bar{\mathbf{S}}(\xi_j,\eta_k).$$

The term $\Delta A(\xi_j,\eta_k)$ is the discretized area at the evaluated point $\bar{\mathbf{S}}(\xi_j,\eta_k)$, which is the area of the quadrangle determine by the four vertices $\bar{\mathbf{S}}(\xi_j,\eta_k), \bar{\mathbf{S}}(\xi_{j+1},\eta_k), \bar{\mathbf{S}}(\xi_j,\eta_{k+1}), \bar{\mathbf{S}}(\xi_{j+1},\eta_{k+1})$. The quadrangle area is computed as the sum of the two triangles.

Again, it can be demonstrated that the approximate continuous formulation II (30) for the surface case is equivalent to the discrete form (2) and (7a) by a scale of $1/n_P$ under the assumption of uniform sampling of the mean shape.

Integration by Gaussian quadrature. The only difference in the Gaussian integration of continuous formulation II compared with that of continuous formulation I is the added the Jacobian term and its normalizer. Because the Jacobian term $|\mathbf{J}|$ is not a polynomial over the parameter domain due to the square root norm, the analytical form cannot be derived for the formulation II ; however, the analytical integrand can be obtained in the Gauss integration.

4. Covariance matrix of a shape population under reparameterization

4.1. Reparameterization via B-splines $\mathbf{R}(\mathbf{u})$

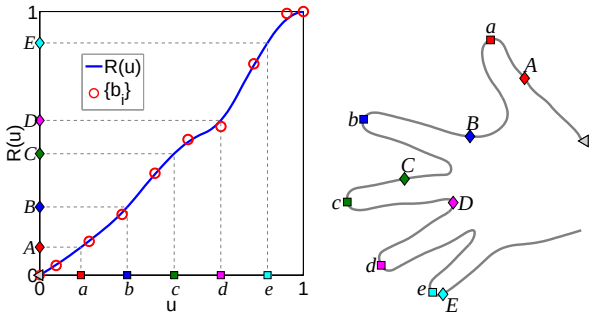
In computing the continuous formulations of the covariance matrix of a shape population, either analytically or approximately, we have adopted the B-spline representation $\mathbf{S}(\mathbf{u})$ of the shapes. To study the influence of shape parameterization (i.e. how points are sampled or distributed) on the covariance matrix, we present a method below for computing the covariance matrix of shapes after B-spline reparameterization, i.e., $\mathbf{S}[\mathbf{R}(\mathbf{u})]$ where $\mathbf{R}(\mathbf{u})$ is the reparameterization function represented again in B-splines. Such reparameterization is also used to optimize correspondence across the shape population in the next section.

4.1.1. Reparameterization of curves

Reparameterization function in the curve case $\mathbf{R}(\mathbf{u})$ could be directly represented by a degree d B-spline function as

$$R(u) = \sum_{i=0}^{n_B} N_i^d(u) b_i, \qquad 0 \leqslant u \leqslant 1, \tag{31}$$

where N_i^d is the B-spline basis function associated with the i-th reparameterization B-spline coefficient b_i defined on a non-decreasing knot vector $U = \{\bar{u}_0, \bar{u}_1, ..., \bar{u}_{n_B+d+1}\}$. The boundary of the parameter domain can be fixed by using a clamped knot vector and setting $b_0 = 0,\ b_{n_B} = 1$ so that $R(0) = 0$ and $R(1) = 1$.



(a) Reparameterization function $R(u)$ represented by quadratic B-spline of 10 coefficients

(b) Points a, b, c, d, e moved to A, B, C, D, E after reparameterization

Figure 2: Direct reparameterization of a B-spline curve

Figure 2(a) displays a reparameterization function expressed by a quadratic reparameterization B-spline with 10 B-spline coefficients b_i. The effect of the reparameterization applied to the parameter domain can be observed for five points, which move from a, b, c, d, e to A, B, C, D, E, respectively. After reparameterization, the parameters are mapped from $u = \{0.15, 0.33, 0.51, 0.68, 0.86\}$ to $R(u) = \{0.11, 0.29, 0.46, 0.58, 0.88\}$, and their images in the physical domain are also relocated somewhere else, as seen in a B-spline represented finger tip in Figure 2(b).

To define a valid reparameterization that is free of self-intersection, the bijectivity constraint for diffeomorphic reparameterization must be enforced by setting $\mathrm{d}\,R(u)/\mathrm{d}\,u > 0$. Because $R(u)$ is a degree d B-spline function, its derivative is a degree $(d-1)$ B-spline function with $b_{i+1} - b_i$ as B-spline coefficients [18]; we thus have the following explicit constraint for ensuring the diffeomorphic reparameterization of curves

$$b_i - b_{i+1} < 0, \qquad i = 0, 1, ..., n_B - 1. \tag{32}$$

4.1.2. Reparameterization of surfaces

The reparameterization function $\mathbf{R}(\mathbf{u})$ for surfaces is a vector field throughout the square parameter domain with two components $[R_u(u, v), R_v(u, v)]$. It could be directly represented by a degree (d, e) B-spline surface controlled by $(n_{B_u} + 1) \times (n_{B_v} + 1)$ control grid with the definition

$$\mathbf{R}(\mathbf{u}) = \sum_{i=0}^{n_{B_u}} \sum_{j=0}^{n_{B_v}} N_i^d(u) N_j^e(v) \mathbf{b}_{i,j}, \quad 0 \leqslant u, v \leqslant 1, \tag{33}$$

where N_i^d and N_j^e are the B-spline basis functions of degree p and q associated with the (i, j)-th B-spline coefficient 2-tuple $\mathbf{b}_{i,j} = (b_{i,j}^u, b_{i,j}^v)$. They are defined on two sets of non-decreasing knot vectors, $U = \{\bar{u}_0, \bar{u}_1, ..., \bar{u}_{n_{B_u}+d+1}\}$ and $V = \{\bar{v}_0, \bar{v}_1, ..., \bar{v}_{n_{B_v}+e+1}\}$, respectively.

For a fixed boundary at the four corners and four sides, two knot vectors are chosen to be of clamped type and the B-spline coefficients at the four boundaries are either 0 or 1. The bijectivity of reparameterization for the purpose of avoiding self-intersection can be guaranteed by the positivity of the reparameterization Jacobian throughout the parameter domain, i.e.

$$\mathcal{J}(\mathbf{u}) = \begin{vmatrix} \frac{\partial R_u(u,v)}{\partial u} & \frac{\partial R_u(u,v)}{\partial v} \\ \frac{\partial R_v(u,v)}{\partial u} & \frac{\partial R_v(u,v)}{\partial v} \end{vmatrix} > 0, \quad \forall (u, v) \in [0, 1]. \tag{34}$$

Note $\mathcal{J}(\mathbf{u})$ is the Jacobian of the reparameterization mapping $\mathbf{R}(\mathbf{u})$. It is different from the Jacobian of the B-spline shapes $\mathbf{J}(\mathbf{u})$ used in the covariance matrix in the continuous formulation II.

4.2. Incorporation of reparameterization into the covariance matrix

Incorporating reparameterization B-splines into the covariance matrix for both continuous formulations is achieved by replacing the parameter $\mathbf{u}$ with the reparameterization function $\mathbf{R}(\mathbf{u})$.

4.2.1. Continuous formulation I

With the reparameterization $\mathbf{R}_i$ for each shape $\mathbf{S}_i(\mathbf{u})$, the basic form of the covariance matrix in continuous formulation I in (10) becomes

$$C_{i_1 i_2}^I \doteq \frac{1}{n_S - 1} \int_{\mathcal{U}} \widehat{\mathbf{S}}_{i_1}[\mathbf{R}_{i_1}(\mathbf{u})]^T \widehat{\mathbf{S}}_{i_2}[\mathbf{R}_{i_2}(\mathbf{u})] \, \mathrm{d}\mathbf{u}, \tag{35}$$

where

$$\widehat{\mathbf{S}}_i[\mathbf{R}_i(\mathbf{u})] = \mathbf{S}_i[\mathbf{R}_i(\mathbf{u})] - \bar{\mathbf{S}}(\mathbf{R}(\mathbf{u})), \tag{36}$$

and the *continuous mean shape* (11) becomes

$$\bar{\mathbf{S}}(\mathbf{R}(\mathbf{u})) \doteq \frac{1}{n_S} \sum_{i=1}^{n_S} \mathbf{S}_i[\mathbf{R}_i(\mathbf{u})]. \tag{37}$$

Bézier curves under B-spline reparameterization. When the curve shapes are represented by Bézier with B-spline reparameterization, we can still derive the analytical integral after the reparameterization. The reason is that the composition of degree p Bernstein polynomial in a Bézier curve with a degree d piecewise polynomial in a B-spline reparameterization function simply leads to a degree pd piecewise polynomial.

Suppose a degree d reparameterization B-spline has its knots dividing the parameter domain $[0, 1]$ into n_K

spans, i.e., $[\tilde{\xi}_{d+s-1}, \tilde{\xi}_{d+s})(s = 1, ..., n_K)$, only over each span is it possible to make use of the function composition properties of Berstein polynomials for the analytical derivation of the covariance matrix entry.

The original i-th Bézier curve is equivalent to a piecewise degree-p Bézier curve defined on these n_K spans, where the curve on the s-th span is defined by

$$\mathbf{S}_{i;s}(u) = \sum_{j_S=0}^{p} B_{j_S}^{p}(\hat{u})\mathbf{P}_{j_S}^{(i;s)}, \tag{38}$$

and the original reparameterization B-spline for the i-th shape is equivalent to a piecewise degree-d Bézier

$$R_{i;s}(u) = \sum_{j_R=0}^{d} B_{j_R}^{d}(\hat{u})\ b_{j_R}^{(i;s)}, \tag{39}$$

over the knot span of $u \in [\tilde{\xi}_{d+s-1}, \tilde{\xi}_{d+s})$ where

$$\hat{u} = \frac{u - \tilde{\xi}_{p+s-1}}{\tilde{\xi}_{p+s} - \tilde{\xi}_{p+s-1}} \in [0, 1], \qquad s = 1, ..., n_K.$$

The i-th Bézier shape after reparameterization over the s-th span is just a function composition of degree-p Bézier $\mathbf{S}_{i;s}$ and degree-d Bézier $R_{i;s}$, yielding a degree-pd Bézier due to the properties of Bernstein basis composition as suggested by [17]

$$\mathbf{S}_{i;s}[R_{i;s}(u)] = \sum_{j=0}^{pd}(1 - \hat{u})^{pd-j}\hat{u}^{j}\ \widetilde{\mathbf{Y}}_{j}^{(i;s)}, \tag{40}$$

where $\widetilde{\mathbf{Y}}_j$ is a sum of the scaled Bernstein coefficients. Releasing the binomial coefficients in (40) restores the Bernstein basis

$$\mathbf{S}_{i;s}[R_{i;s}(u)] = \sum_{j=0}^{pd} B_{j}^{pd}(\hat{u})\ \mathbf{Y}_{j}^{(i;s)}, \text{s.t. } \mathbf{Y}_{j}^{(i;s)} = \widetilde{\mathbf{Y}}_{j}^{(i;s)} \Big/ \binom{pd}{j} \tag{41}$$

The mean shape is still a degree-pd Bézier curve

$$\bar{\mathbf{S}}_s(u) = \sum_{j=0}^{pd} B_{j}^{pd}(\hat{u})\bar{\mathbf{Y}}_{j}^{(s)}, \text{where: } \bar{\mathbf{Y}}_{j}^{(s)} = \frac{1}{n_S}\sum_{i=1}^{n_S}\mathbf{Y}_{j}^{(i;s)}, \tag{42}$$

and the mean-removed shape for the i-th shape is also a Bézier curve

$$\hat{\mathbf{S}}_{i;s}[R_{i;s}(u)] = \mathbf{S}_{i;s}[R_{i;s}(u)] - \bar{\mathbf{S}}_s(u) = \sum_{j=0}^{pd} B_{j}^{pd}(\hat{u})\mathbf{Z}_{j}^{(i;s)}, \tag{43}$$

with control points: $\mathbf{Z}_{j}^{(i;s)} = \mathbf{Y}_{j}^{(i;s)} - \bar{\mathbf{Y}}_{j}^{(s)}$. The covariance matrix entry is

$$\begin{aligned} C_{i_1 i_2}^{I} &= \int_0^1 \frac{1}{n_S - 1}\hat{\mathbf{S}}_{i_1;s}[R_{i_1;s}(u)]^T\ \hat{\mathbf{S}}_{i_2;s}[R_{i_2;s}(u)]\ \mathrm{d}u \\ &= \frac{\sum_{s=1}^{n_K}\left(\tilde{\xi}_{p+s} - \tilde{\xi}_{p+s-1}\right)\sum_{j=0}^{2pd} W_{j}^{(i_1,i_2;s)}}{(n_S - 1)(2pd + 1)}, \end{aligned} \tag{44}$$

where $W_{j}^{(i_1,i_2;s)} = \sum_{l=\max(0,j-pd)}^{\min(j,pd)} \frac{\binom{pd}{l}\binom{pd}{j-l}}{\binom{2pd}{j}}\ \mathbf{Z}_{l}^{(i_1;s)T}\mathbf{Z}_{j-l}^{(i_2;s)}$.

B-spline curves under B-spline reparameterization. B-spline curves generally have more than one knot span; an inverse map must be computed to locate those knot spans after reparameterization because the B-spline basis functions are a single polynomial only within each knot span of the B-spline. For any diffeomorphic reparameterization $R(u)$, there exists a unique set of parameters $\Psi = \{\psi_0, \psi_1, ..., \psi_{n+p+1}\}$ that are mapped to the training set B-spline knots Ξ, namely,

$$R(\psi_k) = \xi_k \qquad k = 0, 1, ..., n + p + 1. \tag{45}$$

where ξ_k is the knots of the B-spline curves.

The inversely mapped training set B-spline knots Ψ with the reparameterization B-spline knots together divide the parameter domain $[0, 1]$ into a new set of spans, over each of which the function composition property of Bernstein polynomial can be applied and (40) holds. The covariance matrix can be analytically computed by (44) over the new set of spans.

Bézier/B-spline surface under B-spline reparameterization. However, such analytical formulas are not available in the case of B-spline reparameterization for Bézier or B-spline surfaces. The reason is that the reparameterized domain $\mathbf{R}(\mathbf{u})$ is typically not rectangular and B-spline surfaces are only piecewise polynomial in rectangular knot intervals.

In summary, we can evaluate the covariance matrix in continuous formulation I analytically by strictly following the analytical formulas listed in Table 1.

Numerical approximations. After applying the reparameterization function, the required modifications for mid-point and Gaussian integrations are straightforward by substituting $\mathbf{R}(\mathbf{u})$ for $\mathbf{u}$ in all the approximate equations displayed in the previous section.

Table 2 gives the minimum number of quadrature points required for exact recovery of the covariance matrix of continuous formulation I. The first four rows are

Table 1: Analytical formulas for continuous formulation I with/without B-spline reparameterization

Training set shape type	Without reparam.	With reparam.
Bézier curves	(20)	(44)
Bézier surfaces	(22)	—
B-spline curves	(20)	(44)
B-spline surfaces	(22)	—

for splines represented shapes without reparameterization and the last four rows are for those with reparameterization. Note that for shapes represented by B-spline curves with B-spline reparameterization (7-th row in Table 1), an inverse mapping is needed for exact recovery of the covariance matrix. For B-spline surfaces with multiple knots upon B-spline reparameterization (8-th row in Table 1), Gauss quadrature cannot guarantee exact recovery of the covariance matrix.

Table 2: Minimal Gaussian abscissae number per knot span n_G^* for exact recovery of the covariance matrix of continuous formulation I

Shape representation $\{\mathbf{S}_i(u)\}$		Reparam. B-spline $\{R_i(u)\}$	Gauss Pt. Number
Type	Degree	Degree	n_G^*
Bézier curve	p	—	$p+1$
Bézier surface	$p \times q$	—	$(p+1)(q+1)$
B-spline curve	p	—	$p+1$
B-spline surface	$p \times q$	—	$(p+1)(q+1)$
Bézier curve	p	d	$pd+1$
Bézier surface	$p \times q$	$d \times e$	$(pd+1)(qe+1)$
B-spline curve	p	d	$pd+1$
B-spline surface	$p \times q$	$d \times e$	—

4.2.2. Continuous formulation II

With reparameterization of the shapes, the basic form (12) of the covariance matrix in continuous formulation II becomes

$$C_{i_1 i_2}^{II} \doteq \frac{\int_{\mathcal{U}} \widehat{\mathbf{S}}_{i_1}[\mathbf{R}_{i_1}(\mathbf{u})]^T \widehat{\mathbf{S}}_{i_2}[\mathbf{R}_{i_2}(\mathbf{u})]\, |\mathbf{J}(\mathbf{R}(\mathbf{u}))|\, \mathrm{d}\mathbf{u}}{(n_S - 1)\int_{\mathcal{U}} |\mathbf{J}(\mathbf{R}(\mathbf{u}))|\, \mathrm{d}\mathbf{u}} \tag{46}$$

where the *continuous mean shape* (11) becomes (37), $\mathbf{J}(\mathbf{R}(\mathbf{u}))$ is computed from the mean shape, and $\widehat{\mathbf{S}}_i[\mathbf{R}_i(\mathbf{u})]$ is defined as in (36).

The analytical integrand can be similarly achieved with the procedure described in Section 3.2.1. This analytical integrand can then be used in the mid-point or Gauss quadrature method for integrating entry in the covariance matrix.

5. Shape Correspondence Optimization via Reparameterization

With the B-spline representation of reparameterization functions $\mathbf{R}(\mathbf{u})$ and the diffeomorphic conditions (32) and (34) presented in Section 4.1.1 and 4.1.2, , we thus have the following optimization formulation for using B-spline based reparameterization for manipulating shape correspondence in B-spline curves/surfaces:

$$\min_{\mathbf{b}} \sum_{\lambda_i \geq \lambda_{\text{cut}}} \left[1 + \log \frac{\lambda_i(\mathbf{b})}{\lambda_{\text{cut}}}\right] + \sum_{\lambda_i < \lambda_{\text{cut}}} \frac{\lambda_i(\mathbf{b})}{\lambda_{\text{cut}}} \tag{47a}$$

$$\text{s.t.} \quad \mathbf{C}(\mathbf{b})\, \mathbf{v}_i(\mathbf{b}) = \lambda_i(\mathbf{b})\mathbf{v}_i(\mathbf{b}) \tag{47b}$$

$$\mathbf{v}_i^T(\mathbf{b})\mathbf{v}_i(\mathbf{b}) = 1, \quad i = 1, ..., n_S \tag{47c}$$

$$g(\mathbf{b}) < 0 \tag{47d}$$

In this formulation, $\mathbf{b}$ is the set of optimization variables and represents the collection of interior B-spline coefficient tuples $\mathbf{b}$ for B-spline reparameterization of $n_s - 1$ shapes. The objective function $f(\mathbf{b})$ is the simplified description length as proposed in [19], which is a function of eigenvalues computed from (47b) and (47c). The covariance matrix $\mathbf{C} = \frac{1}{n_S - 1}\hat{\mathbf{C}}$ could be computed by analytical form or numerical schemes (mid-point or Gauss quadrature). The constraint (47d) represents the diffeomorphic conditions, i.e. (32) for curves and (34) for surfaces, each of which is a function of optimization variables $\mathbf{b}$.

6. Numerical examples

In this section, we compare the numerical results of computed covariance matrices from two continuous formulations under different discretization resolutions. To compare the analytical form of the covariance matrix (10) from continuous formulations and their approximations through mid-point or Gauss quadrature and their convergence, we compare the covariance matrix norm and its largest eigenvalue. The matrix norm used is the Frobenius norm of an $m \times n$ matrix $\mathbf{C} = \{C_{ij}\}$ $(i = 1, ..., m; j = 1, ..., n)$

$$|\mathbf{C}| = \sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n} C_{ij}^2}. \tag{48}$$

We demonstrate the convergence of both formulations, but first formulation leads to different covariance matrix norms under different reparameterizations. We then compare the resulting shape correspondence in building the SSM.

6.1. Covariance matrices from discrete points and spline representation of shapes

To compare the computed covariance matrices from two forms of shape representations: discrete points and B-spline based continuous representation, we compare the computed matrices for a simple set of shapes: four circular arcs, as indicated in Figure 3(a). These four quarter circle shapes can be represented with the following parametric form

$$\begin{aligned}
\mathbf{S}_1(u) &= \left[\sin\left(\frac{\pi}{2}u\right), \cos\left(\frac{\pi}{2}u\right)\right] \\
\mathbf{S}_2(u) &= \left[1-\cos\left(\frac{\pi}{2}u\right), \sin\left(\frac{\pi}{2}u\right)\right] \\
\mathbf{S}_3(u) &= \left[1-\cos\left(\frac{\pi}{2}u\right), 1-\sin\left(\frac{\pi}{2}u\right)\right] \\
\mathbf{S}_4(u) &= \left[\sin\left(\frac{\pi}{2}u\right), 1-\cos\left(\frac{\pi}{2}u\right)\right]
\end{aligned} \tag{49}$$

where $u \in [0, 1]$. With such explicit, continuous representations of arcs, the covariance matrices of continuous formulation I (10) and formulation II (12) can be directly computed, without resort to discrete sampling or B-spline fitting. The resulting matrix norms are 0.1284 for formulation I and 0.1200 for formulation II. We then compute the two forms of covariance matrices from the sampled discrete points and B-spline representations of sampled points. It should be noted that the non-rational form of B-splines used in this work can only approximate the circular arcs. Exact representation of a circular arc would need a rational form of B-splines.

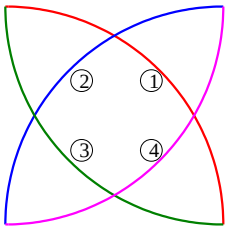
We choose three forms of point sampling on the shapes (Figure 3(b)): uniform sampling based on the angle span θ, uniform sampling along x axis, and uniform sampling along each chord of the arcs. The span-angle-based uniform sampling also corresponds to arc-length based uniform sampling because the underling shapes are circular. Figure 3(c), (d) and (e), respectively, indicate the sampled points based on the three sampling schemes for which the number of sampled points $n_P = 16$. The norm of the discrete covariance matrix $\frac{1}{n_P}\widetilde{D}_{i_1 i_2}$ using the discrete formula (7a) is indicated in Figure 3(f), which the x-axis indicates the number of sampled points that ranges from 6 to 10^4. The norm of another weighted discrete covariance matrix (28) with the substitution of sampled points $\mathbf{x}_i^{(j)}$ for $\mathbf{S}_i(\xi_j)$ is displayed in Figure 3(g). It can be seen that, for each sampling scheme, different numbers of sampled points correspond to different covariance matrix norms. As the number of sampled points increases, the covariance matrix norm of each sampling scheme converges, but each converges to different values, depending on the underling sample scheme. For all three sampling schemes, it takes at least 10^3 sampled points to converge. The norms of the discrete covariance matrix under the angle span, X-coordinate, and chordal distance sampling schemes converge to 0.1284, 0.1309 and 0.1200 respectively as indicated in Figure 3(f). When using weighted points in the covariance matrix; they converge to 0.1200, 0.1310 and 0.1200 as indicated in Figure 3(g). Among the three point sampling schemes, only angle span based sampling has a converged value consistent with reference value 0.1284 because it reflects the arc-length parameterization. This example clearly demonstrates that, for the covariance matrix from the discrete-point-based representation of shapes, a large number of data points is required to reach the converged covariance matrix and the resulting matrix depends on the sampling density (number of points) and distribution (sampling scheme).

For the B-spline based covariance matrix, we first fit B-spline curves based on the discrete points sampled with the above three schemes. B-spline curve/surface fitting usually involves three phases [20], knot determination whereby a set of knot parameters for B-spline shapes need to be determined, data parameterization whereby for each data point Q_j a corresponding parameter $\bar{u}_j$ needs to be decided, and control point calculation whereby control points P_i are computed so that the error between the data and the resulting B-spline shape is minimized, i.e.,

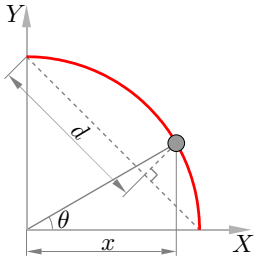
$$\min_{\{\mathbf{P}_i\}} \quad \sum_{j=1}^{n_P} |\mathbf{Q}_j - \mathbf{S}(\bar{u}_j)|^2.$$

We choose uniform distribution of knots with clamped end conditions. For data parameters, we studied three common forms of data parameterization displayed in Figure 4: a) equidistant $\Delta\bar{u}_j$ = constant; b) chord length $\Delta\bar{u}_j = \|\Delta Q_j\|$; and c) the centripetal method $\Delta\bar{u}_i = \sqrt{\|\Delta Q_{j+1} - Q_j\|}$ [21] where Q_j represents the data point.

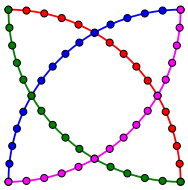
Figure 5(a) indicates an example of fitting n_P =6 sampled points with a quadratic B-spline of $n_{CP} = 4$ control points, and Figure 5(b) is the fitted B-spine training set. In the convergence study of formulation I (computed by the analytical formula (20)) and formulation II computed by Gaussian quadrature approximation of (12), we choose $n_P = 100$ sampled points for B-spline



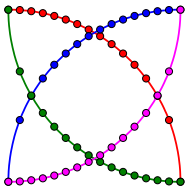
(a) Four quarter circles



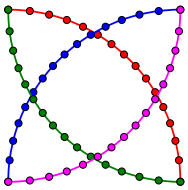
(b) Three sampling schemes: angle θ, x-coordinate, and chord d



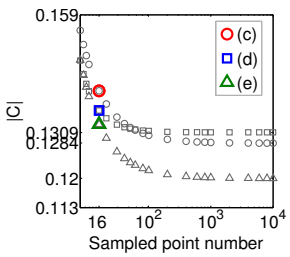
(c) Angle span sampling



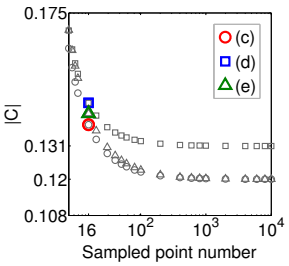
(d) X-coordinate sampling



(e) Chordal distance sampling

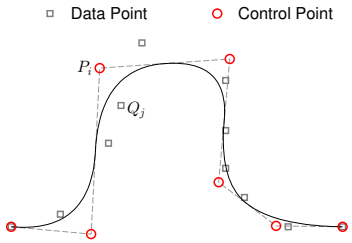


(f) Covariance matrix norm from discrete points



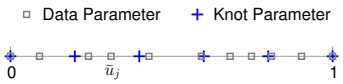
(g) Covariance matrix norm from weighted discrete points

Figure 3: Discrete sampling and corresponding covariance matrix norms. The sign (circle, square and triangle) in Figures 3(f) and 3(g) corresponds to the point sampling from the schemes based on angle span, x-coordinate and chord distance, respectively.



(a) B-spline fitting from discrete data

(b) Equidistant data parameterization: $\Delta\bar{u}_j$ = constant



(c) Chord length data parameterization: $\Delta\bar{u}_j \sim \|\Delta Q_{j+1} - Q_j\|$

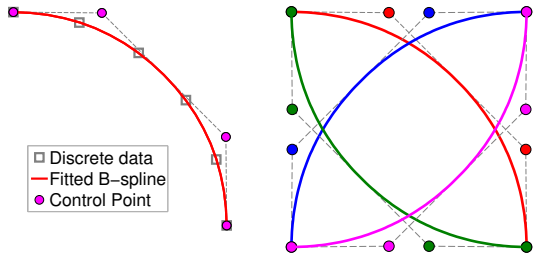
(d) Centripetal data parameterization: $\Delta\bar{u}_j \sim \sqrt{\|\Delta Q_{j+1} - Q_j\|}$

Figure 4: B-spline fitting with different methods of data parametrization.

curve fitting and vary the number of control points n_{CP} from 4 to 20. The results of the two continuous forms of covariance matrices computed with fitted B-splines curves are displayed in the remainder of Figure 5, in which the three data parameterization methods for three sampling schemes are displayed in the last three rows in Figure 5. We can see from this figure that

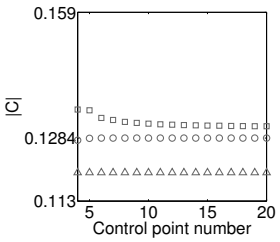
- Although covariance matrices from discrete points all need at least a thousand points to converge as indicated in Figure 3, the convergence of the covariance matrix from B-splines only needs fewer than 10 control points as indicated in Figure 5. Thus, B-spline based shape representation is very efficient for computing the covariance matrix.

- With the chord-length based data parameterization in B-spline fitting, the resulting covariance matrices for both forms of continuous formulation are independent from point sampling schemes, as indicated in Figure 5(e) and 5(f). This is because, when a sufficient number of data points are used in fitting, regardless of sampling schemes, the chord-length-based data parameterization essentially corresponds to arc length-based data parameterization. Thus, B-spline-based shape represen-

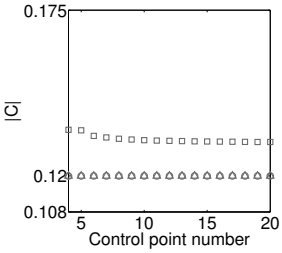


(a) Fitted B-spline curve

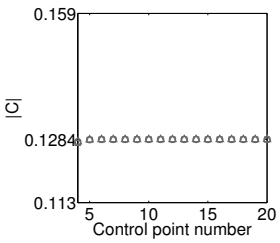
(b) Fitted B-spline training set



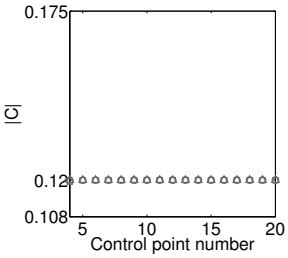
(c) Equidistant data parameterization, formulation 1



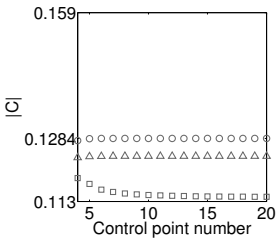
(d) Equidistant data parameterization, formulation 2



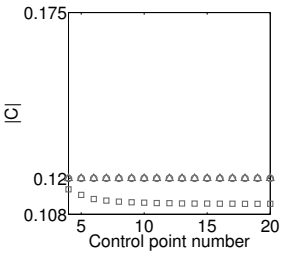
(e) Chord length based data *parameterization* , formulation 1



(f) Chord length based data parameterization, formulation 2



(g) Centripetal data *parameterization* , formulation 1



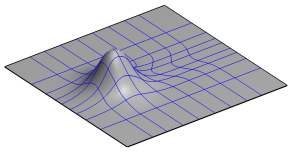
(h) Centripetal data *parameterization* , formulation 2

Figure 5: Covariance matrix norms from B-spline curves fitted with three methods of data parameterization. The sign (circle, square and triangle) corresponds to respectively angle span, x-coordinate and chord distance based point sampling scheme shown in Figures 3.

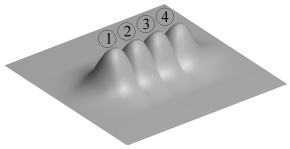
tation under chord length-based data parameterization can lead to accurate calculation of the covariance matrix, regardless of point sampling schemes. However, for other forms of data parameterization such as equidistant and centripetal methods, the B-spline-based covariance matrix still depends on the data parameterization method. The reason is that any data parameterization other than the chord-length-based method would create a correspondence among shapes that is different from that under arc-length-based parameterization. Thus, the resulting covariance matrix may differ depending on the underlying data parameterization.

6.2. *Convergence of approximated continuous formulations*

A plane-bump shape instance is represented by a bi-quadratic B-spline surface by 11×11 control points as indicated in Figure 6(a) where the knot curves are displayed in blue. The training set is composed of 4 such B-spline surfaces, which differ in the horizontal position of the bump along the u-direction as seen in Figure 6(b).

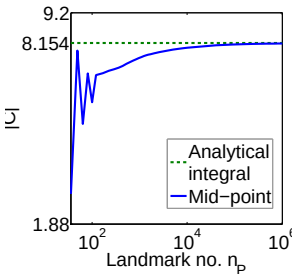


(a) Shape in bi-qaudratic B-spline surface

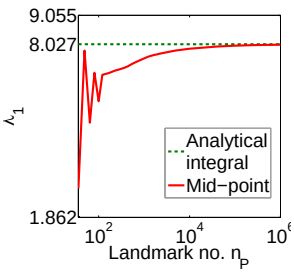


(b) 4 B-spline surfaces superimposed

Figure 6: Four B-spline represented plane-bumps superimposed. $n_S = 4; p = 2, q = 2$.



(a) Covariance matrix norm

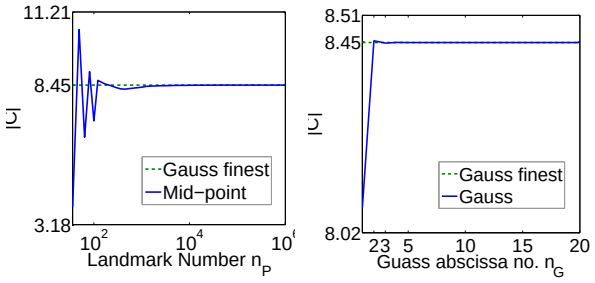


(b) Largest eigenvalue

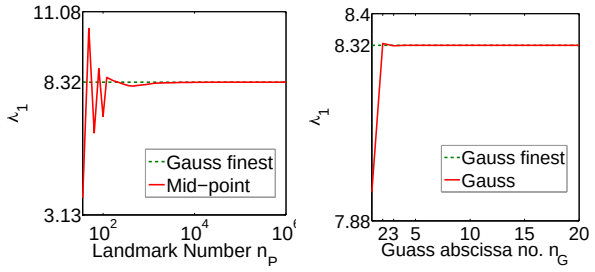
Figure 7: Convergence of covariance matrix norm and eigenvalues with continuous formulation I and its approximation by the mid-point scheme for the plane-bump training set shown in Fig. 6.

In the approximated continuous formulation I through the mid-point, $n_P = n_{Pu} \times n_{Pv} = n_{Pu}^2$ land-

marks are sampled on each B-spline surface throughout the parameter domain $\mathcal{U} = [0,1]\times[0,1]$; and a series of landmark numbers for n_{Pu} along both u- and v-direction ranging from $10 \sim 10^3$ are used. The analytical covariance matrix is computed following the derived equation (22). The matrix norm of the analytical covariance matrix is $|\mathbf{C}^*| = 8.15$ and the largest analytical eigenvalue is $\lambda_1^* = 8.03$. It can be seen from Figure 7 that both the covariance matrix and the eigenvalues under the approximate formulation are approaching the analytical values as the number of landmarks tend toward infinity. In addition, the analytical values can also be efficiently and exactly obtained by integration with Gauss quadrature, where only $n_{Gu}^* = n_{Gv}^* = 3$ Gauss abscissae are needed per knot span in each direction.



(a) Covariance matrix norm (b) Covariance matrix norm



(c) Largest eigenvalue (d) Largest eigenvalue

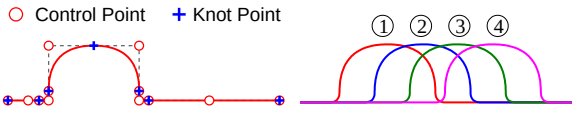
Figure 8: Convergence of the covariance matrix norm and eigenvalues with continuous formulation II and its approximations for the plane-bump training set shown in Fig. 6.

In the mid-point based continuous formulation II (30), $n_P = n_{Pu} \times n_{Pv} = n_{Pu}^2$ landmarks are sampled on each B-spline surface throughout the parameter domain $\mathcal{U} = [0,1]\times[0,1]$; and a series of landmark numbers for n_{Pu} along both u- and v-direction ranging from $10 \sim 10^3$ are used. The converged covariance matrix is computed with the Gauss quadrature ($n_G = 20$) and the matrix norm of the most accurate covariance matrix is $|\mathbf{C}^*| = 8.45$ and the largest analytical eigenvalue is $\lambda_1^* = 8.32$. Figure 8 indicates that the covariance matrix and the eigenvalues are computed with a different number of mid-points and Gauss quadrature points. They all converge as the number of landmarks tend toward infinity or the number of quadrature points increase. It is clear that the matrix norm and the eigenvalues can be more efficiently and accurately obtained with Gauss quadrature than with mid-point.

Comparing Figure 7 and Figure 8, one can see that approximations of both formulations I and II converge as sufficient landmark points are used. However, they converge to different values, 8.154 and 8.45 respectively for the matrix norm, and 8.027 and 8.32 respectively for the largest eigenvalue.

6.3. Convergence of continuous formulations under reparameterization

In this example, we indicate that, with the first continuous formulation, the converged values may be different with different parameterizations and the second formulation is parameterization-independent. In the training set of line-bumps, each line-bump shape is represented by a quadratic B-spline governed by 12 control points as indicated in Figure 9(a). The training set then consists of 4 such B-spline curves, with only the horizontal position of the bump different, as displayed in Figure 9(b). This synthetic training set is simply the open curve version of the benchmark box-bump as used in [22], with the bump feature further highlighted.

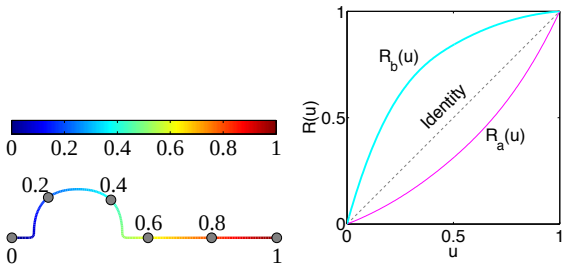


(a) Shape in quadratic B-spline of 12 control points (b) 4 B-splines curves superimposed

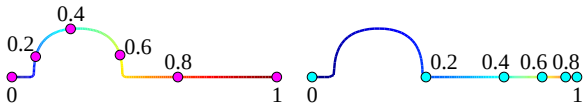
Figure 9: Line-bump training set represented as B-spline curves. $n_S = 4$; $p = 2$.

The reparameterization functions are indicated in Figure 10 where the color field of Figure 10(a) signifies the original parameterization of Shape 1. It is also indicated as the identity function in Figure 10(b). Two reparameterization functions $R_a(u)$ and $R_b(u)$ in Figure 10(b) are used to generate different parameterizations in Figure 10(c) and Figure 10(d), respectively. $R_a(u)$ curved downward with a slight deviation from the identity function makes the parameterization slightly squeezed toward $u = 0$; $R_b(u)$ curved upward with a large deviation from the identity function makes the parameterization severely squeezed toward $u = 1$.

We applied the three reparameterization functions to all B-spline represented line-bumps. Figures 11 and

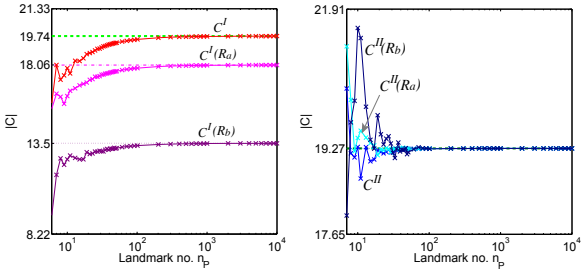


(a) Original parameterization (b) Different reparameterizations



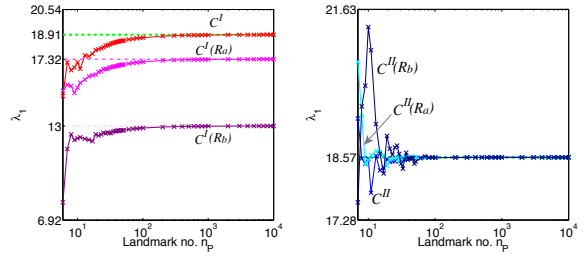
(c) Reparameterization $R_a(u)$ (d) Reparameterization $R_b(u)$

Figure 10: Different reparameterizations of line-bump shapes.



(a) Formulation I (b) Formulation II

Figure 11: Covariance matrix norm $|C|$ under continuous formulation I (C^I) and formulation II (C^{II}) with mid-point integration under three parameterizations for the line-bump training set shown in Fig. 9.



(a) Formulation I (b) Formulation II

Figure 12: Largest eigenvalue λ_1 under continuous formulation I (C^I) and formulation II (C^{II}) with mid-point integration under three parameterizations for the line-bump training set shown in Fig. 9.

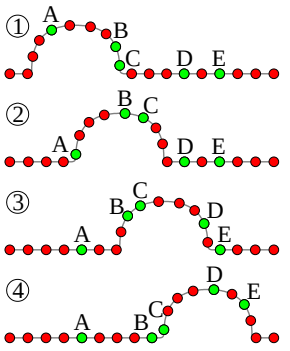
Figure 12 indicate the covariance matrix norm and the largest eigenvalues, respectively, of the line-bump under the three different reparameterizations for both formulations,. The dotted straight lines in Figure 11(a) and Figure 12(a) indicating the convergence limit for continuous formulation I are computed by the analytical formula (44); the dotted straight lines in Figure 11(b) and Figure 12(b) are computed by using $n_G = 20$ Gauss quadrature points per knot span. Figure 11(a) indicates, with continuous formulation I, the matrix norms converge to different values with the original parameterization and reparameterization $R_a(u)$, and $R_b(u)$. On the other hand, with continuous formulation II as indicated in Figure 11(b), the matrix norms all converge to the same value with three different reparameterizations. Similar behaviors can be observed for the largest eigenvalue λ_1 as indicated in Figure 12. This indicates that continuous formulation II is parameterization independent and characterizes the intrinsic geometric property of shapes.

6.4. Optimizing shape correspondence

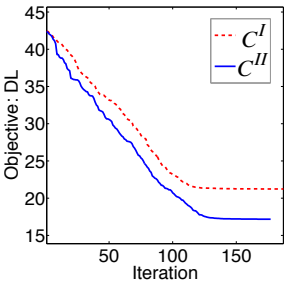
We extend the two continuous formulations with mid-point based integration to optimization of shape correspondence to examine how the landmark resolution and distribution in the usual discrete formulation would affect the correspondence optimization. It is found that at a sufficiently dense resolution of landmarks, the difference between the two formulation approximations does not lead to significant disparities in the resulting shape correspondence. However, when the number of landmarks n_P drops to a certain level, the two formulations often lead to entirely different results in the optimized group-wise correspondence and/or landmark distribution. We present two examples below that illustrate the continuous formulation II advantage over formulation I in terms of correspondence quality and faithful representation of the shapes.

6.4.1. Line-bump

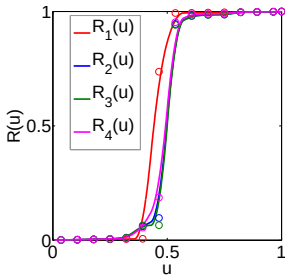
For the line-bump training set consisting of $n_S = 4$ B-spline curves, $n_P = 19$ landmarks are used to represent each shape instance and $n_b = 16$ control coefficients are used to model reparameterization B-spline for each shape instance. The initial landmark configuration is displayed in Figure 13(a), and the five picked (the 5, 9, 10, 14, 16-th) landmarks A,B,C,D,E highlighted in green indicate a poor initial correspondence in which bump corners do not correspond across the training set.



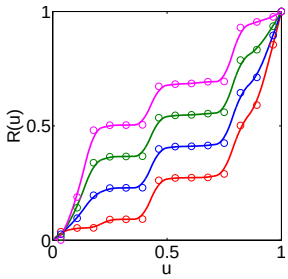
(a) Initial $n_p = 19$ landmarks sampled on each shape



(b) Optimization history of description length with continuous formulation I and formulation II

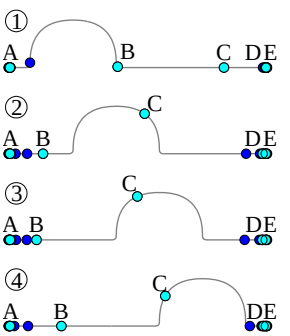


(c) Optimized reparameterization functions using C^I discretization: almost flat on two ends

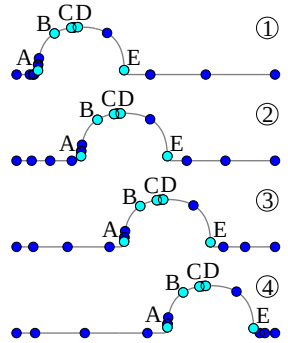


(d) Optimized reparameterization functions using C^{II} discretization: reasonably distributed

Figure 13: Shape correspondence optimization with the covariance matrix of continuous formulation I (C^I) and formulation II (C^{II}).

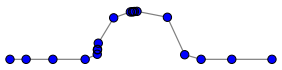


(a) Optimized landmark distributions using C^I approximation



(b) Optimized landmark distributions using C^{II} approximation

(c) Landmark represented mean shape after optimization with C^I



(d) Landmark represented mean shape after optimization with C^{II}

Figure 14: The optimized correspondence from continuous formulation I (C^I) and formulation II (C^{II}).

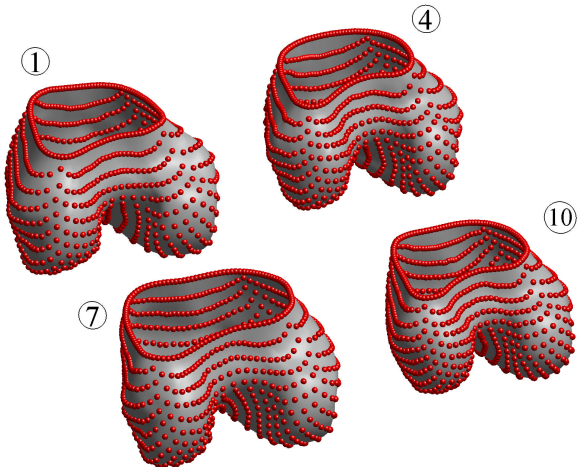
Correspondence optimization is conducted with the covariance matrix computed by both continuous formulation I and II approximations. The optimization history of DL objective function is shown overlapped in Figure 13(b), and the optimized reparameterization functions $\{R_i(u)\}$ for formulation I and II are displayed in Figure 13(c) and (d) respectively. It is observed that formulation I causes a nearly flat platform near the two ends while formulation II does not; this suggests that formulation I could lead to a severe collapse of landmarks during correspondence optimization. The collapsed landmarks near the two ends after correspondence optimization for formulation I are displayed in Figure 14(a), in which there are barely any landmarks used for representing the bump and the optimized correspondence has very few improvements either as revealed by the locations of the five feature landmarks. On the other hand, the optimized landmark result in Figure 14(b) generated by formulation II gives a remarkably enhanced correspondence as demonstrated by the correspondence of the five feature landmarks A,B,C,D,E. Moreover, the optimized landmarks form a more reasonable distribution and shape representation than that of formulation I.

To further shed light on the differences between the two formulations, the mean shapes consisting of optimized landmarks are also plotted in Figure 14(c) and (d) for formulation I and II, respectively. Recall that as a result of the difference between (10) and (12) regarding the Jacobian term $|\mathbf{J}(\mathbf{u})|$, the associated approximate formulations (26) and (28) differ only in the length weight term $\Delta L(\xi_j)$. The main part in the integrand, $\left[\mathbf{S}_{i_1}(\xi_j) - \bar{\mathbf{S}}(\xi_j)\right]^T \left[\mathbf{S}_{i_2}(\xi_j) - \bar{\mathbf{S}}(\xi_j)\right]$, are the same; and it is just the landmarks in a shape instance minus the mean shape landmarks. In the second formulation, the collapsing of landmarks around point M as indicated in Figure 14(c) would have led to larger weight in $\Delta L(\xi_j)$ for landmark M than that in the first formulation. Thus, the second formulation has the effect of alleviating landmark collapse during correspondence optimization.

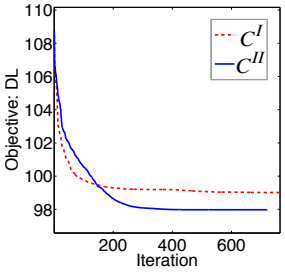
6.4.2. *Distal femur*

In this 3D real training set case of distal femurs, it has $n_S = 10$ B-spline surfaces, each of which is represented by a bi-quadratic B-spline surface with 30×30 control points. $n_P = 31 \times 31$ landmarks are used to represent each shape instance and $n_b = 8 \times 8$ control coefficients are used to model reparameterization B-spline for each shape instance.

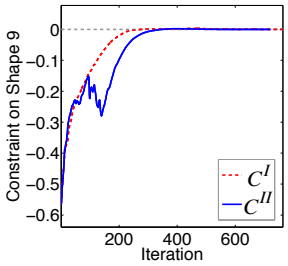
The initial landmark configuration is displayed in Figure 15(a). The optimization history of DL objec-



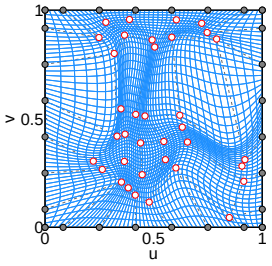
(a) Initial $n_p = 31 \times 31$ landmarks before optimization



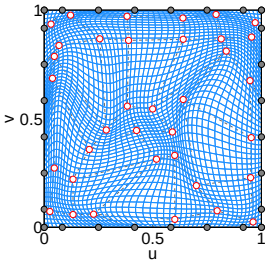
(b) Optimization history of description length for continuous formulation I and formulation II



(c) Constraint history of Shape 9 for continuous formulation I and formulation II

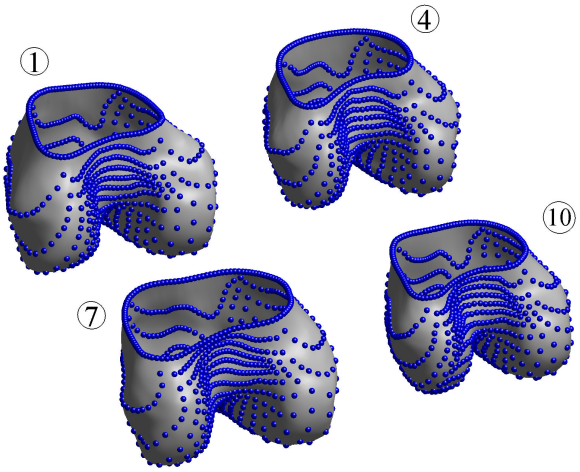


(d) Optimized reparameterization function vector field using C^I formulation

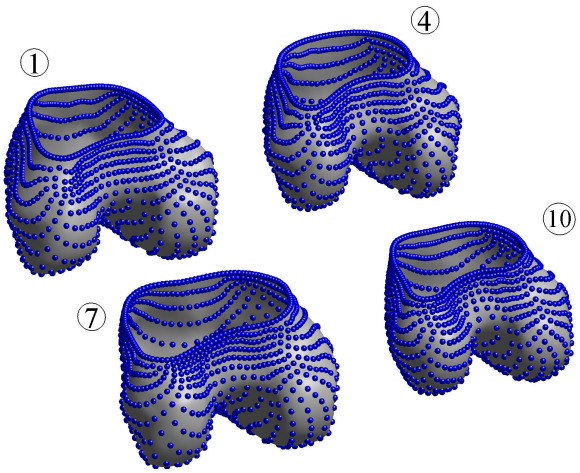


(e) Optimized reparameterization function vector field using C^{II} formulation

Figure 15: Correspondence optimization for 10 femur bones with the covariance matrix of formulation I and II.



(a) Optimized landmarks using approximation of formulation I: improved correspondence as indicated by reduced DL objective function, but with rather insufficient landmarks for shape representation in the curved regions on the two sides.



(b) Optimized landmarks using C^{II} approximation: improved correspondence as indicated by reduced DL objective function, and with improved correspondence and with sufficient and reasonably distributed landmarks for shape representation in the two curved regions.

Figure 16: The difference between the mid-point integration of continuous formulation I (C^I) and formulation II (C^{II}) for computing the covariance matrix can lead to substantial difference in the resulting shape correspondence after the optimization.

tive function for both formulations is overlapped in Figure 15(b). The constraint history of Shape 9, $g_9(u)$ for enforcing the positivity of Jacobian $\mathcal{J}(\mathbf{u})$ in (34), is shown in Figure 15(b), where optimized constraints become active. The optimized reparameterization function $\{R_i(u)\}$ parametric grids for formulation I and II are displayed in Figure 15(d) and (e) respectively.

The optimized landmarks for both formulations give comparable correspondence improvements in terms of the three measures for statistical shape models (generalization ability, specificity and compactness [13]). However for formulation I, the optimized landmarks are not distributed in a way to sufficiently represent the underlying shape due to under-sampling in the bump region as indicated in Figure 16(a). In contrast, formulation II not only gives comparable correspondence improvement, but also provides a sufficient shape sampling and reasonable optimized landmarks distribution as indicated in Figure 16(b). This desirable feature of formulation II is from the shape Jacobian term and its approximation, i.e. the length/area weight associated with landmarks, which incorporates the effect of geometric variation from each shape instances into the whole training set's statistical shape variation.

7. Conclusion

In this paper, we have presented methods for accurately and efficiently computing continuous formulations of the covariance matrix in which B-splines are used both as a shape representation and as a form of reparameterization. We have indicated, with B-spline representation of the shapes, the formulation I is amenable to analytical computing without sampling or discretization. Numerical approaches based on midpoint and Gauss quadrature are developed for approximating both continuous formulations. We have indicated that the first formulation is parameterization-dependent, i.e. it may lead to different covariance matrices with different parameterizations, and the second formulation is parameterization-independent. We have demonstrated that the proposed closed-form and numerical procedure for computing the covariance matrix are both accurate and efficient in the sense that it would take many more discrete points in the usual discrete form of the covariance matrix to converge to the same matrix. We have also indicated that, when data points are parameterized with the chord length method in fitting B-spline curves, the resulting covariance matrix does not depend on the point sampling scheme.

With B-spline representations, both formulations have been successfully applied in correspondence optimization for minimizing description length of the statistical shape model. Our numerical results demonstrate, with sufficient sampling, both formulations lead to similar shape correspondence. When the sampling is not sufficient, the second formulation is more robust in alleviating the potential collapse of landmarks.

Our approach is based on B-spline based parametric curves and surfaces. How to extend it to shapes of complex topology is a challenge for future research.

Acknowledgments

This work is supported in part by AFOSR grant FA9550-12-1-0206 and NSF grant 0900597.

References

[1] T. F. Cootes, C. J. Taylor, D. H. Cooper, J. Graham, et al., Active shape models-their training and application, Computer vision and image understanding 61 (1) (1995) 38–59.

[2] T. Heimann, H.-P. Meinzer, et al., Statistical shape models for 3d medical image segmentation: A review, Medical image analysis 13 (4) (2009) 543.

[3] T. F. Cootes, G. J. Edwards, C. J. Taylor, Active appearance models, Pattern Analysis and Machine Intelligence, IEEE Transactions on 23 (6) (2001) 681–685.

[4] N. Hasler, C. Stoll, M. Sunkel, B. Rosenhahn, H.-P. Seidel, A statistical model of human pose and body shape, in: Computer Graphics Forum, Vol. 28, Wiley Online Library, 2009, pp. 337–346.

[5] P. P. Smyth, C. J. Taylor, J. E. Adams, Automatic measurement of vertebral shape using active shape models, Image and Vision Computing 15 (8) (1997) 575–581.

[6] K.-K. Shen, J. Fripp, F. Mériaudeau, G. Chételat, O. Salvado, P. Bourgeat, Detecting global and local hippocampal shape changes in Alzheimer's disease using statistical shape models, Neuroimage 59 (3) (2012) 2155–2166.

[7] R. R. Paulsen, Statistical shape analysis of the human ear canal with application to in-the-ear hearing aid design, IMM, Informatik og Matematisk Modellering, Danmarks Tekniske Universitet, 2004.

[8] F. Deligianni, A. J. Chung, G.-Z. Yang, Nonrigid 2-d/3-d registration for patient specific bronchoscopy simulation with statistical shape modeling: Phantom validation, Medical Imaging, IEEE Transactions on 25 (11) (2006) 1462–1471.

[9] C.-H. Chu, Y.-T. Tsai, C. C. Wang, T.-H. Kwok, Exemplar-based statistical model for semantic parametric design of human body, Computers in Industry 61 (6) (2010) 541–549.

[10] T. F. Cootes, C. J. Taylor, D. H. Cooper, J. Graham, Training models of shape from sets of examples, in: BMVC92, Springer, 1992, pp. 9–18.

[11] A. Kotcheff, C. J. Taylor, et al., Automatic construction of eigenshape models by direct optimization., Medical Image Analysis 2 (4) (1998) 303.

[12] R. H. Davies, C. J. Twining, P. Daniel Allen, T. F. Cootes, C. J. Taylor, Building optimal 2d statistical shape models, Image and Vision Computing 21 (13) (2003) 1171–1182.

[13] R. Davies, C. Twining, C. J. Taylor, Statistical models of shape: Optimisation and evaluation, Springerverlag London, 2008.

[14] J. Hladuvka, K. Bühler, MDL spline models: Gradient and polynomial reparameterisations, in: 17th British Machine Vision Conference, 2005, pp. 869–878.
[15] R. H. Davies, C. J. Twining, T. F. Cootes, C. J. Taylor, Building 3-d statistical shape models by direct optimization, Medical Imaging, IEEE Transactions on 29 (4) (2010) 961–981.
[16] I. Jolliffe, Principal component analysis, Wiley Online Library, 2005.
[17] R. T. Farouki, V. Rajan, Algorithms for polynomials in Bernstein form, Computer Aided Geometric Design 5 (1) (1988) 1–26.
[18] G. E. Farin, Curves and surfaces for CAGD: a practical guide, Morgan Kaufmann, 2002.
[19] H. H. Thodberg, Minimum description length shape and appearance models, in: Information Processing in Medical Imaging, Springer, 2003, pp. 51–62.
[20] L. A. Piegl, W. Tiller, The NURBS book, Springer Verlag, 1997.
[21] E. T. Lee, Choosing nodes in parametric curve interpolation, Computer-Aided Design 21 (6) (1989) 363–370.
[22] R. H. Davies, C. J. Twining, T. F. Cootes, J. C. Waterton, C. J. Taylor, A minimum description length approach to statistical shape modeling, Medical Imaging, IEEE Transactions on 21 (5) (2002) 525–537.