DIRECT DIFFEOMORPHIC REPARAMETERIZATION FOR CORRESPONDENCE OPTIMIZATION IN STATISTICAL SHAPE MODELING

BY

KANG LI

Submitted in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy in Mechanical Engineering
in the Graduate College of the
Illinois Institute of Technology

Approved ________________________
Advisor

Chicago, Illinois
May 2015

ACKNOWLEDGMENT

First and foremost, I would like to extend my great thanks to my academic advisor Dr. Xiaoping Qian. I joined his research lab with very little knowledge and skills, and soon I am about to walk out with the expertise and ability to do research in a wide range of fields related to computer aided design and manufacturing. This wouldn't have been possible if not for his passion, insights and ethics, every bit of which has inspired, enlightened and guided me on the path to a successful career. His patience, kindness and support have helped me survive the hard times when my thesis topic encountered unexpected setbacks. I am also grateful that he has been funding me throughout my doctoral studies.

I would like to thank the rest of the thesis committee, Dr. Gady Agam, Dr. Kevin Cassel, Dr. Sudhakar Nair and Dr. Boris Pervan for giving me valuable advices and feedbacks from my doctoral comprehensive exam to my thesis defense. My gratitude also goes to the MMAE department for providing all kinds of facilities and opportunities in the teaching, research and networking, from which I have benefited a great deal as a graduate student.

I would like to acknowledge my senior fellow lab mate Pinghai Yang, for his generous and warmhearted help with courses and programming during my very first year in the lab. Besides, I take great delight in working with all the other fellow lab mates, Wei Zhao, Mingming Wang, Kangmin Xu, Arash Kalantari, Songtao Xia, Yu Liu, Xilu Wang, Noah Jaxon, Seungcheol Yang, Jing Li, Tian Chen, Long Chen, Cunfu Wang and Yaqi Zhang, all of whom have made the intensive and challenging Ph.D. life easier and merrier.

I am deeply grateful for the financial support of AFOSR grant FA9550-12-1-0206 and NSF grants #0900597 and #1030347. As regards my publications, I am greatly thankful to Active Appearance Models group at Technical University of

Denmark for the hand data; to Dr. Alejandro A Espinoza Orías at Rush University Medical Center for the femoral bone data; to Dr. Wei Sun and Tissue Mechanics Lab at Georgia Institute of Technology for the aortic data. A sincere thank you to Dr. Rhondri Davies at the University of Manchester UK, Dr. Hans Henrik Thodberg at BoneXpert, Dr. Martin Styner at the University of North Carolina - Chapel Hill, for the useful comments and suggestions in the communication.

Finally, my fullest gratitude goes to my parents and my sister Yao for standing by my side all along; and I would also like to give my special thanks to my wonderful wife Yuanlin for being there with me through the difficult times during my doctoral study.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

This dissertation proposes an efficient optimization approach for obtaining shape correspondence across a group of objects for statistical shape modeling. With each shape represented in a B-spline based parametric form, the correspondence across the shape population is cast as an issue of seeking a reparametrization for each shape so that a quality measure of the resulting shape correspondence across the group is optimized. The quality measure is the description length of covariance matrix of the shape population, with landmarks sampled on each shape. The movement of landmarks on each B-spline shape is controlled by the reparameterization of the B-spline shape. The reparameterization itself is also represented with B-splines and B-spline coefficients are used as optimization parameters. We have developed formulations for ensuring the bijectivity of the reparameterization. A gradient-based optimization approach is developed, including techniques such as constraint aggregation and adjoint senstivity for efficient, direct diffeomorphic reparameterization of landmarks to improve the group-wise shape correspondence. Numerical experiments on both synthetic and real 2D and 3D data sets demonstrate the efficiency and effectiveness of the proposed approach.

CHAPTER 1
INTRODUCTION

1.1 Introduction

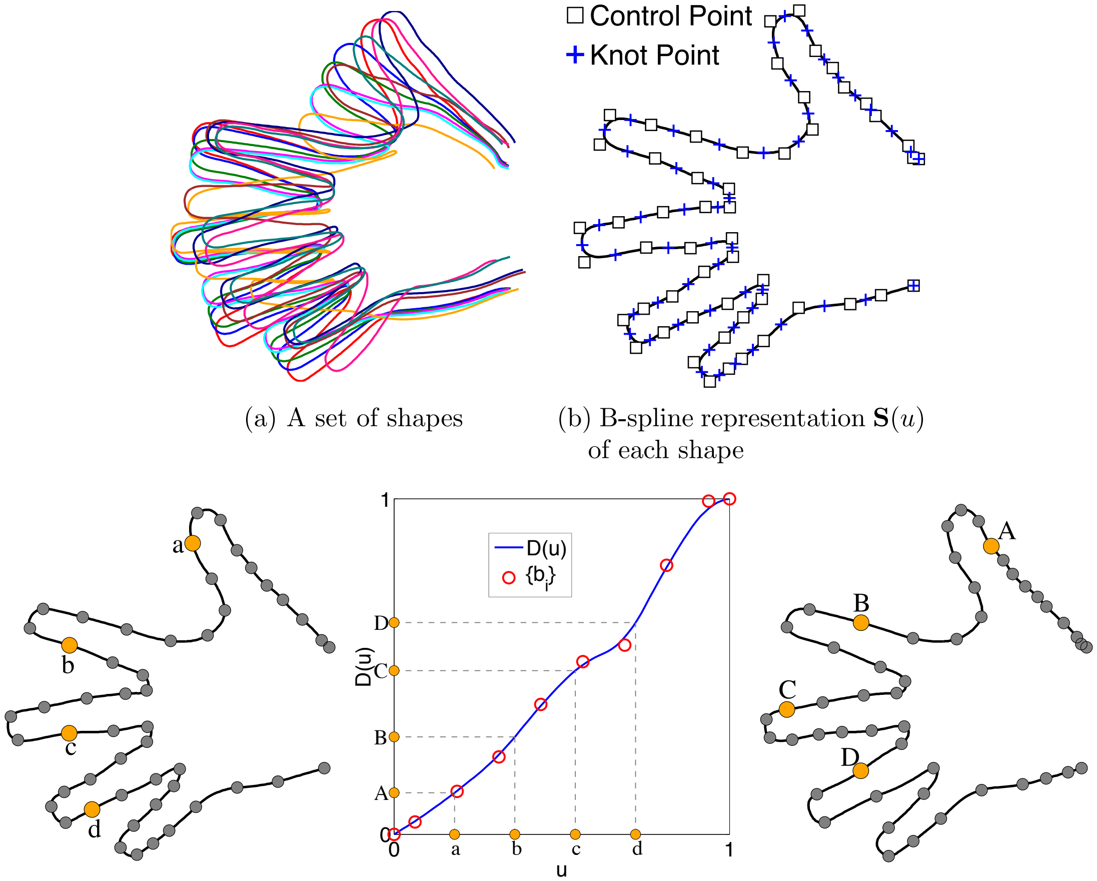
Statistical shape model (SSM) provides a compact characterization of the shape variability in a set of shapes. It was initially used as a tool for facilitating automatic image segmentation [1, 2]. It has since seen many other applications including facial recognition [3], computer animation [4], medical diagnosis [5, 6] and patient-specific modeling [7, 8, 9, 10], to name but a few. Finding shape correspondence across all shape instances is a fundamental task in building SSM, and the quality of shape correspondence directly affects the quality of the resulting SSM. Manual identification of landmarks is effective under some circumstances but in general is not a reliable strategy since it tends to be subjective, time-consuming, error prone, and difficult to be applied in large scale data sets [11]. Consequently, methods for automatically identifying the shape correspondence have been a major research focus in the field.

The automatic identification of the shape correspondence across a set of objects can be achieved by either “template-based” or “population-based” correspondence optimization. The former finds one by one the pairwise correspondence between a pre-selected template shape and each shape instance in the training set, and the found pairwise correspondences are then propagated through the common template reference to form the group-wise correspondence; the latter seeks to directly find the groupwise correspondence across all shape instances simultaneously. The SSM quality of the template-based approach is inherently subject to bias from the template selection, whereas the population-based approach considers the entire training set and tends to give more reliable SSMs. Generally both template-based and population-based approaches search correspondence by formulating it as an optimization problem and minimizing a certain objective function. Various correspondence search techniques have been compared and analyzed in [12]. The pairwise search usually reduces to a problem where a “similarity” measure between the template and each shape instance is minimized and some regularization constraints are satisfied. Typically such measure is related to geometric descriptors such as spatial distance and shape feature, and the optimization is essentially a rigid or non-rigid registration problem [13, 14, 15, 16]. The population based search directly optimizes a quality measure of the statistical model obtained from all the instances. During the past few years, SSM quality measures have evolved from the model covariance trace [17], to the model covariance determinant [18], and finally to the *Description Length*(DL) [19, 20, 21] and its simplification [22] or variants [23]. This information theoretic objective function of description length has shown to be an effective measure [19] for the population-based correspondence optimization.

Although the population-based approach to shape correspondence does not require the pre-selection of a template and tends to provide a more faithful characterization of the variability pattern, this approach is still far from being widely used to build SSMs due to its low efficiency in identifying optimal correspondence across the shape population. In the minimum description length based group-wise correspondence optimization approach originated in [19], the group-wise shape correspondence search consists of successive small-scale optimizations, each of which uses only a few optimization variables to relocate landmarks in a local region of each shape instance. In each optimization, only landmarks in local regions are moved. This necessitates a huge number of successive optimizations to manipulate all the landmarks, thus leading to inefficiency. Some researchers use analytical gradient formula whenever possible to speed up the gradient evaluation [24, 25]. However, in these implementations, the landmark positions in the training set shapes are non-differentiable with respect to optimization variables, the gradients are thus only partially analytical. In [26], spline representation of 2D shapes is proposed so a full analytical gradient of reparameterization can be derived.

In our proposed approach, we cast shape correspondence as an issue of seeking optimal reparameterization $\mathbf{D}(\mathbf{u})$ of the parametric field $\mathbf{u}$ of each shape so that a quality measure f of the resulting shape correspondence across a group of objects is optimized. The reparameterization is applied to the parametric domain of parameterized curves or surfaces. Our SSM is based on the point-distribution model [27]. In our approach, each landmark point $\mathbf{S}(\mathbf{u})$ in a given shape is changed to $\mathbf{S}\left(\mathbf{D}(\mathbf{u})\right)$ in order to improve correspondence via the reparameterization $\mathbf{D}(\mathbf{u})$. Our approach thus requires the parameterization of each shape, that is, every point $\mathbf{x}$ of the shape in the physical space is mapped to a point $\mathbf{u}$ in the parametric domain. In our implementation, we choose the B-spline representation $\boldsymbol{S}(\mathbf{u})$ of each shape instance, which can be reconstructed from triangular mesh representation of 3D objects. The parametric domain then undergoes a reparameterization represented via another tensor-product B-spline $\mathbf{D}(\mathbf{u})$ with B-spline coefficients $\mathbf{b}$ as the optimization parameters. We choose the description length as the objective function of the shape correspondence.

Figure 1.1 illustrates the proposed idea. A group of hand contours are shown in Figure 1.1(a). Each shape is represented with B-splines, and Figure 1.1(b) shows such a B-spline representation for one shape with control points and knots. Initially landmarks are uniformly sampled over the parameter domain of the B-spline shape $\mathbf{S}(u)$ as shown in Figure 1.1(c). To change the landmark positions, reparameterization $\mathbf{D}(u)$ is applied to the parameter domain of each B-spline shape. This reparameterization is also represented with B-splines as shown in in Figure 1.1(d) where each red point represents a B-spline coefficient for the reparameterization. The landmarks are redistributed as shown in Figure 1.1(e) after the reparametrization. The landmark redistribution can be seen from the four highlighted landmarks, where a, b, c, d moved to A, B, C, D respectively over the other side of the finger tips.



(a) A set of shapes

(b) B-spline representation $\mathbf{S}(u)$ of each shape

(c) Landmarks sampled on $\mathbf{S}(u)$

(d) Reparameterization function $\mathbf{D}(u)$

(e) Landmarks $\mathbf{S}(\mathbf{D}(u))$ after reparameterization

Figure 1.1. Landmark manipulation by reparameterization for improving shape correspondence in statistical shape modeling.

The salient feature of this approach is as follows

- *Diffeomorphic deformation through B-splines.* The reparametrization could be achieved by concatenating a seres of simple homeomorphic mappings as in [21]. One optimization run with this reparametrization technique leads to the deformation of a local parametric region cannot provide any information on the search direction for subsequent optimizations. Therefore it requires the concatenation of a large number of simple mappings and causes severe inefficiency (See Section 5.1). Instead of concatenation, we propose the use of single B-spline functions to directly represent the diffeomorphic reparameterization $\mathbf{D}(\mathbf{u})$ for the parameterization $\mathbf{u}$ of each shape instance $\mathbf{S}(\mathbf{u})$. The parametric field is reparameterized with B-spline functions $\mathbf{D}(\mathbf{u})$ where injectivity for the reparameterization is guaranteed by enforcing the Jacobian positivity constraint.
- *Full differentiability of the objective function f (i.e. description length) with respect to reparameterization variables* $\mathbf{b}$. The objective function f (i.e. description length) is a function of landmark positions. The landmark positions in each shape are differentiable with respect to reparameterization parameters $\mathbf{b}$ due to the parametric representation $\mathbf{S}(\mathbf{u})$ of each shape and diffeomorphic reparameterization $\mathbf{D}(\mathbf{u})$. This ensures that the description length is fully differentiable with respect to the reparameterization variables $\mathbf{b}$.

The direct diffeomorphic reparameterization based formulation for SSM leads to an optimization problem with a large number of constraints (for enforcing the injectivity of reparametrization) and a large number of optimization variables (i.e. B-spline coefficients for reparameterization). Due to the full differentiability of the objective function f (i.e. description length) with respect to reparameterization variables $\mathbf{b}$, a gradient-based optimization approach can be developed to ensure fast convergence. More specifically, the following optimization techniques are developed.

- *Constraint aggregation.* The B-spline based diffeomorphic reparameterization leads to a large number of constraints on Jacobians for ensuring the mapping is bijective. In order to facilitate fast convergence in gradient-based optimization, a constraint aggregation technique is used where the large number of constraints are aggregated into one or a few constraints.

- *Adjoint method for computing sensitivity.* The adjoint approach is used to compute the sensitivity of the objective function with respect to reparameterization parameters $\mathbf{b}$, which is more efficient than direct differentiation of the objective function f. In computing the sensitivity of the description length w.r.t. optimization variables $\boldsymbol{b}$, eigenvalues of the covariance matrix and their derivatives are needed. Since each eigen analysis is expensive, the adjoint method is thus especially efficient for computing the sensitivity in this kind of optimization problems that have larger number of optimization variables and fewer functions (after constraint aggregation). In this adjoint method, the derivatives of a function w.r.t. a large number of optimization variables only involves one eigen analysis of the covariance matrix. On the other hand, in the direct differentiation method, the number of eigen analysis is the same as the number of optimization variables.

1.2 Background on SSM and covariance matrix

Previously we have pointed out the research motivation for the ultimate goal of building statistical models, and briefly described the basic idea of the proposed method. In order for a rigorous discussion, we lay out the necessary mathematical contexts that are already known to the community. More specifically, the detailed computing procedures of statistical shape modeling will be presented, and two types of covariance matrix formulations will be identified with both the discrete and continuous shape representations.

1.2.1 Background on statistical shape modeling. Statistical Shape Model was initially called *Point Distrubition Model* (PDM) [27], and it requires each shape instance in the training set be represented by a set of points, known as the *landmarks* [17]. Regardless of the geometric form of the training set, landmarks are constrained to be on the boundary of a shape instance, and they form a point-based representation that approximates the original shape [28]. The statistical modeling framework requires that the same number of landmarks sampled on all shapes across the training set.

Suppose a training set $\{\mathbb{T}_i\}$ $(i = 1, 2, \cdots, n_S)$ comprises n_S shape instances and each is represented by n_P landmarks. The j-th landmark of the i-th instance is $\mathbf{x}_i^{(j)} = [x^{(j)}, y^{(j)}, z^{(j)}]_i^T \in \mathbb{R}^3$. Owing to the correspondence assumption of landmarks, all the n_S landmarks $\{\mathbf{x}_i^{(j)}\}$ $(i = 1, 2, \cdots, n_S)$ with label j should correspond across all instances. The landmark representation of each instance is usually written into a concatenation of n_P landmarks ordered by labels as a *shape vector* expression:

$$\mathbf{x}_i \doteq [x_i^{(1)}, y_i^{(1)}, z_i^{(1)}, x_i^{(2)}, y_i^{(2)}, z_i^{(2)}, \cdots, x_i^{(n_P)}, y_i^{(n_P)}, z_i^{(n_P)}]^T.$$

All the n_S shape vectors $\{\mathbf{x}_i\}$ could be concatenated into a $3n_P \times n_S$ shape vector matrix:

$$\mathbf{X}^S \doteq [\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_{n_S}]. \tag{1.1}$$

In order to align a group of shapes stored in $\mathbf{X}^S$, the *Generalized Procrustes Analysis* (GPA) [29] operation is conducted, denoted by an alignment operator as below

$$\mathbf{X}^A = \mathcal{A}(\mathbf{X}^S). \tag{1.2}$$

Specifically, the GPA of group-wise alignment is done by iteratively performing the pair-wise Procrustes Analysis (PA) between each shape and the mean. The PA brings a shape vector $\mathbf{x}$ to a fixed shape vector $\mathbf{y}$ by similarity transformation

$$\min_{\mathbf{t},s,\mathbf{R}} ||\mathbf{y} - s\mathbf{R}\left(\mathbf{x} - \mathbf{t}\right)||^2 \doteq \sum_{j=1}^{n_P} \left|\left|\mathbf{y}^{(j)} - s\mathbf{R}\left(\mathbf{x}^{(j)} - \mathbf{t}\right)\right|\right|^2$$

where shape irrelevant factors including translation $\mathbf{t}$, scaling s and rotation $\mathbf{R}$ are removed. For more details in the iterative procedure, refer to Algorithm 2.1 in [30].

If the training shapes are continuous and the i-th shape is parameterized by the mapping $\mathbf{S_i}(\mathbf{u})$, the continuous representation of the covariance matrix expressed in the (μ, ν)-th entry is

$$\mathbf{E}_{\mu\nu} = \frac{1}{(n_S - 1)A} \int \left[\mathbf{S}_\mu(\mathbf{u}) - \bar{\mathbf{S}}(\mathbf{u})\right] \cdot \left[\mathbf{S}_\nu(\mathbf{u}) - \bar{\mathbf{S}}(\mathbf{u})\right] \mathrm{d}A(\mathbf{u}), \tag{1.3}$$

where $\mathbf{S}(\cdot)$ is the vector-valued function that defines the continuous representation of the i-th shape by mapping the parameter space to the physical space. $\bar{\mathbf{S}}(\mathbf{u})$ is the mean shape and A is the surface area of the mean shape. For numerical implementation, the continuous covariance matrix is obtained via discretization through a set of discrete landmarks as

$$\mathbf{E}_{\mu\nu} = \frac{1}{(n_S - 1)n_P} \sum_{i=1}^{n_S} (\mathbf{x}_i - \bar{\mathbf{x}})_\mu (\mathbf{x}_i - \bar{\mathbf{x}})_\nu. \tag{1.4}$$

where on each shape the i-th landmark $\mathbf{x}_i$ could be obtained as sampling at the i-th parameter point $\mathbf{u}_i$: i.e. $\mathbf{x}_i = \mathbf{S}(\mathbf{u}_i)$

This could be written simply in a matrix form [19]

$$\mathbf{E} = \frac{1}{(n_S - 1)n_P} \mathbf{X}_c^T \mathbf{X}_c, \tag{1.5}$$

where $\mathbf{X}_c$ is defined by

$$\mathbf{X}_c \doteq [\mathbf{x}_1^A - \bar{\mathbf{x}}, \mathbf{x}_2^A - \bar{\mathbf{x}}, \cdots, \mathbf{x}_{n_S}^A - \bar{\mathbf{x}}], \tag{1.6}$$

and $\mathbf{x}_i^A$ is the i-th shape vector after alignment operation $\boldsymbol{\mathcal{A}}$, i.e. the component of $\mathbf{X}^A$ in (1.2) ; the *mean shape vector* of SSM is

$$\bar{\mathbf{x}} = \frac{1}{n_S} \sum_{i=1}^{n_S} \mathbf{x}_i^A. \tag{1.7}$$

The *Principal Component Analysis* (PCA) [31] is then conducted to extract the principal modes of shape variability via the eigenvalue decomposition of the covariance matrix

$$\mathbf{E}\mathbf{v}_m = \lambda_m \mathbf{v}_m \quad (m = 1, 2, \cdots, n_S - 1), \tag{1.8}$$

where $\mathbf{v}_m$ is the m-th eigenvector and λ_m the corresponding eigenvalue.

The mean shape $\bar{\mathbf{x}}$, modes $\{\mathbf{v}_m\}$ and variances $\{\lambda_m\}$ constitutes the statistical shape model. This statistical model is a much more compact representation of the shape variability pattern of the implied shape class than the original training set. What's more, it makes possible to represent any valid instance $\mathbf{x}$ belonging to the shape class by a linear approximation using only the first $\widetilde{m}$ ($\widetilde{m} \leq n_S - 1$) modes

$$\mathbf{x} \approx \ \bar{\mathbf{x}} + \sum_{m=1}^{\widetilde{m}} \beta_m \mathbf{v}_m, \tag{1.9}$$

where the m-th mode parameter is found by projection

$$\beta_m = (\mathbf{x} - \bar{\mathbf{x}})^T \mathbf{v}_m. \tag{1.10}$$

The quality of the linear approximation has a great influence on the utility of the statistical model in subsequent applications, and it is evidently decided by the quality of the SSM, which is directly tied to the quality of the groupwise correspondence.

The PCA step defined by (1.5) and (1.8) is compactly written as

$$\boldsymbol{\lambda} = \mathcal{C}(\mathbf{X}^A). \tag{1.11}$$

The objective function of Description Length was originally derived by Davies in [19] and elaborated in [28]. A simplified version presented in [22] defined as below is used in our correspondence optimization:

$$f \doteq \sum_{m=1}^{n_S - 1} L_m, \tag{1.12}$$

where each mode's contribution is

$$L_m = \begin{cases} 1 + \log(\lambda_m / \lambda_{\text{cut}}) & \lambda_m \geq \lambda_{\text{cut}}, \\ \lambda_m / \lambda_{\text{cut}} & \text{otherwise.} \end{cases}$$

The threshold λ_{cut} is determined by landmark resolution and shape scale

$$\lambda_{\text{cut}} = \frac{2\, l_{\min}}{r_{\max}}, \tag{1.13}$$

where $l_{\min}$ is the smallest edge length in the landmark-based representation and $r_{\max}$ is the radius of largest circumscribing sphere over training set shapes.

1.2.2 Background on discrete and continuous formulations of the covariance matrix.

As a critical mathematical entity in statistical modeling stream, the covariance matrix bares significant statistical information and is thus influential on the resultant SSM. It is necessary to distinguish the two types of formulations under different shape representations in terms of the continuity.

1.2.2.1 Discrete formulation.

We first review 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, 28].

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*

$$\mathbf{X}_i \doteq \left[\mathbf{x}_i^{(1)}, \mathbf{x}_i^{(2)}, ..., \mathbf{x}_i^{(n_P)}\right]^T, \tag{1.14}$$

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

$$\mathbf{D} \doteq \frac{1}{n_S - 1} \sum_{i=1}^{n_S} \left(\mathbf{X}_i - \bar{\mathbf{X}}\right) \left(\mathbf{X}_i - \bar{\mathbf{X}}\right)^T, \tag{1.15}$$

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

$$\bar{\mathbf{X}} \doteq \frac{1}{n_S} \sum_{i=1}^{n_S} \mathbf{X}_i. \tag{1.16}$$

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

$$\mathcal{X} = [\mathbf{X}_1 - \bar{\mathbf{X}},\ \mathbf{X}_2 - \bar{\mathbf{X}},\ ...,\ \mathbf{X}_{n_S} - \bar{\mathbf{X}}], \tag{1.17}$$

which is of size $3n_P \times n_S$; the covariance matrix (1.15) could then be compactly expressed by

$$\mathbf{D} = \frac{1}{n_S - 1} \mathcal{X}\mathcal{X}^T. \tag{1.18}$$

The *Principal Component Analysis* (PCA) [31] is frequently used to decompose the covariance matrix into the principal modes of shape variability via eigen-decomposition

$$\mathbf{D}\mathbf{v}_m = \lambda_m \mathbf{v}_m, \tag{1.19}$$

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 covariance matrix definition is the following,

$$\widetilde{\mathbf{D}} \doteq \frac{1}{n_S - 1} \sum_{i=1}^{n_S} \left(\mathbf{X}_i - \bar{\mathbf{X}}\right)^T \left(\mathbf{X}_i - \bar{\mathbf{X}}\right) \tag{1.20a}$$

$$= \frac{1}{n_S - 1} \mathcal{X}^T \mathcal{X}, \tag{1.20b}$$

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

$$\widetilde{D}_{i_1 i_2} \doteq \frac{1}{n_S - 1} \left(\mathbf{X}_{i_1} - \bar{\mathbf{X}}\right)^T \left(\mathbf{X}_{i_2} - \bar{\mathbf{X}}\right), \tag{1.21}$$

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

$$\lambda_m = \widetilde{\lambda}_m, \tag{1.22a}$$

$$\mathbf{v}_m = \mathcal{X}\, \widetilde{\mathbf{v}}_m,\ \widetilde{\mathbf{v}}_m = \mathcal{X}^T \mathbf{v}_m. \tag{1.22b}$$

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 (1.22a) and eigenvectors that can be mutually converted by (1.22b).

This formulation featuring (1.15) or (1.20a) 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 [27] in statistical shape modeling. It is referred to as the "*discrete formulation*" in this thesis.

1.2.2.2 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.

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

$$C^I_{i_1 i_2} \doteq \frac{1}{n_S - 1} \int_{\mathcal{U}} \left[\mathbf{S}_{i_1}(\mathbf{u}) - \bar{\mathbf{S}}(\mathbf{u})\right]^T \left[\mathbf{S}_{i_2}(\mathbf{u}) - \bar{\mathbf{S}}(\mathbf{u})\right]\ \mathrm{d}\mathbf{u}, \tag{1.23}$$

where the mean shape is

$$\bar{\mathbf{S}}(\mathbf{u}) \doteq \frac{1}{n_S} \sum_{i=1}^{n_S} \mathbf{S}_i(\mathbf{u}). \tag{1.24}$$

$\mathbf{C}^I = \{C^I_{i_1 i_2}\}(i_1, i_2 = 1, ..., n_S)$ 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 [18] and is adopted in [26] for cardinal spline curves.

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

$$C^{II}_{i_1 i_2} \doteq \frac{\displaystyle\int_{\mathcal{U}} \left[\mathbf{S}_{i_1}(\mathbf{u}) - \bar{\mathbf{S}}(\mathbf{u})\right]^T \left[\mathbf{S}_{i_2}(\mathbf{u}) - \bar{\mathbf{S}}(\mathbf{u})\right] |\mathbf{J}(\mathbf{u})| \, \mathrm{d}\mathbf{u}}{\displaystyle (n_S - 1) \int_{\mathcal{U}} |\mathbf{J}(\mathbf{u})| \, \mathrm{d}\mathbf{u}}, \tag{1.25}$$

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 [21], and it is equivalent to the "length/area" element $\mathrm{d}A(\mathbf{x})$ in [28]. The difference in the formulation (1.25) 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 2.3.

For curves, the Jacobian term is

$$|\mathbf{J}(u)| = \left|\frac{\mathrm{d}\,\bar{\mathbf{S}}(\mathbf{u})}{\mathrm{d}\,u}\right|, \tag{1.26}$$

and for surfaces, the Jacobian term is

$$|\mathbf{J}(u, v)| = \left|\frac{\partial\,\bar{\mathbf{S}}(u, v)}{\partial u} \times \frac{\partial\,\bar{\mathbf{S}}(u, v)}{\partial v}\right|. \tag{1.27}$$

1.3 Thesis overview

The objective of this thesis is to develop an algorithm that efficiently optimizes shape correspondence for building statistical model. The means is a direct reparametrization technique based on B-spline that makes the correspondence manipulation for a set of shapes in a much more efficiently fashion than current state-of-the-art method.

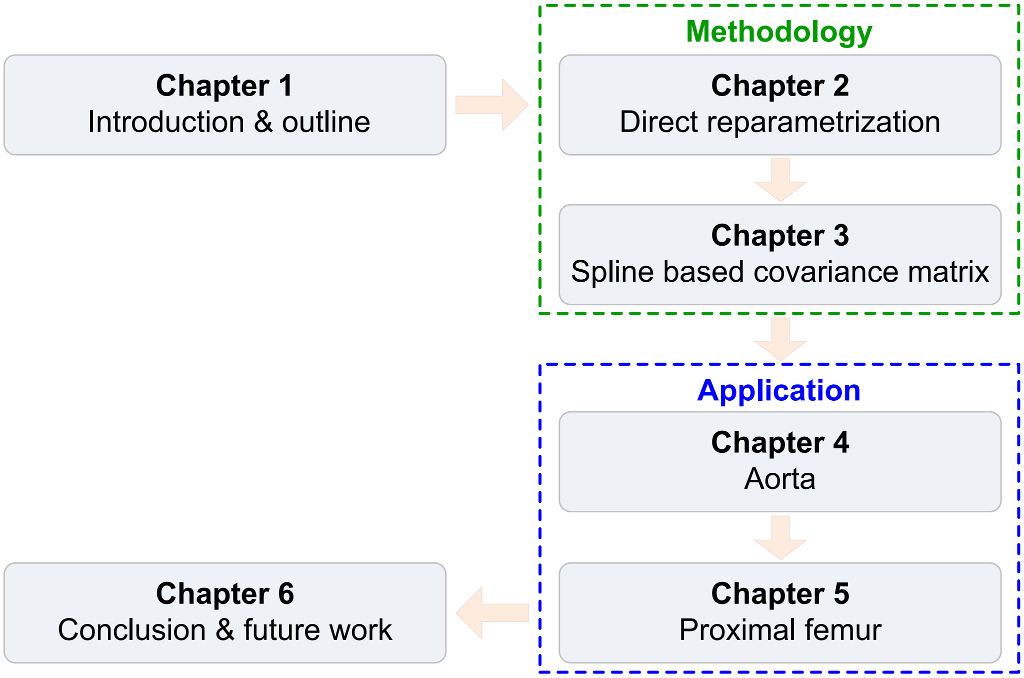


Figure 1.2. Thesis chapters organization

The thesis is organized as shown in Figure 1.2. Chapter 1 overviews the whole thesis; Chapter 2 and Chapter 3 form the key methodology part; Chapter 4 and Chapter 5 demonstrate two applications; Chapter 6 summarizes and concludes. More specific outline goes as follows

- Chapter 1 provides a brief overview of the research field the mathematical basics of statistical shape modeling.
- Chapter 2 describes the core technique of B-spline based direct reparametrization, including its mathematical formulation, incorporation into correspondence optimization and numerical experimentations.
- Chapter 3 elaborates further on the covariance matrix properties in the context of B-spline structure for both shape set representation and reparametriztaion.

- Chapter 4 applies the proposed method to one real medical data of human aorta, which offers meaningful clinical insights to heart disease study.

- Chapter 5 uses the method on a real medical data se of the proximal femur, for the pathological analysis of cam impingement.

- Chapter 6 sums up all the contents, research contributions and points out future directions.

The main chapters of the thesis are based on the publications below

- Chapter 2: Kang Li and Xiaoping Qian. **Direct Diffeomorphic Reparameterization for Correspondence Optimization in Statistical Shape Modeling**. *Computer-Aided Design*, vol. 64, pp. 33-54, 2015.

- Chapter 3: Kang Li and Xiaoping Qian. **Covariance Matrix of A Shape Population: A Tale on Spline Setting**. *Computers & Graphics*, vol. 47, pp. 89-104, 2015.

- Chapter 4: Kang Li, Xiaoping Qian, Caitlin Martin and Wei Sun. **Toward patient-specific computational study of aortic diseases: A population based shape modeling approach**, *Proceedings of the ASME 2014 International Design and Engineering Technical Conferences and Computers and Information in Engineering*, Buffalo, NY, 2014. With Prakash Krishnaswami CAPPD Best Paper Award.

CHAPTER 2

DIRECT DIFFEOMORPHIC REPARAMETERIZATION FOR CORRESPONDENCE OPTIMIZATION

2.1 Direct diffeomorphic reparameterization based on B-splines

In this section, we present how B-splines can be used for representing reparameterization of parametric curves and surfaces and be used for manipulating shape correspondence. We show how such single direct reparameterization function differs from concatenations of multitude of simple mappings into one reparameterization function. We then show constraints for ensuring the B-spline based reparameterization is diffeomorphic.

Since our correspondence manipulation is based on reparameterization of parametric curves and surfaces, the training shapes must be in the form of parametric curves or surfaces. In this thesis, we choose to use B-splines to represent the shapes.

A B-spline curve of degree d is defined by

$$\mathbf{S}(u) = \sum_{k=0}^{n} B_{k,d}\mathbf{P}_k \qquad 0 \leq u \leq 1;$$

where $B_{k,d}$ is the B-spline basis function [32] of degree d associated with the k-th control points $\mathbf{P}_k$ recursively defined on a non-decreasing knot vector $\Xi = \{\bar{\xi}_0, \bar{\xi}_1, \cdots, \bar{\xi}_{n+d+1}\}$

$$\begin{aligned} B_{k,d}(u) &= \frac{(u - \bar{\xi}_k)B_{k,d-1}(u)}{\bar{\xi}_{k+d} - \bar{\xi}_k} + \frac{(\bar{\xi}_{k+d+1} - u)B_{k+1,d-1}(u)}{\bar{\xi}_{k+d+1} - \bar{\xi}_{k+1}}, \\ B_{k,0}(u) &= \begin{cases} 1 & \bar{\xi}_k \leq u \leq \bar{\xi}_{k+1}, \\ 0 & \text{otherwise.} \end{cases} \end{aligned} \tag{2.1}$$

A B-spline surface of degree d and e is defined by

$$\mathbf{S}(u, v) = \sum_{k=0}^{n_u} \sum_{l=0}^{n_v} B_{k,d}B_{l,e}\mathbf{P}_{k,l} \qquad 0 \leq u \leq 1, 0 \leq v \leq 1;$$

where $B_{k,d}$ and $B_{l,e}$ following the basis definition in (2.1) is associated with the (k,l)-th control point $\mathbf{P}_{k,l}$, and the knot vectors along the two parametric directions are $\Xi = \{\bar{\xi}_0, \bar{\xi}_1, \cdots, \bar{\xi}_{n_u+d+1}\}$ and $\mathbf{H} = \{\bar{\eta}_0, \bar{\eta}_1, \cdots, \bar{\eta}_{n_v+e+1}\}$.

If initially the shapes are not in B-spline form, many algorithms exist for shape parametrization [33], and B-spline fitting [32]. In addition to B-spline, there also exist other parametric representations. We prefer B-spline over other options due to: 1) local support property leading to high ability of shape representation and modeling flexibility; 2) purely polynomial basis functions contributing to the superior computational efficiency in gradient computation during the subsequent correspondence optimization.

2.1.1 B-spline based direct reparameterization.

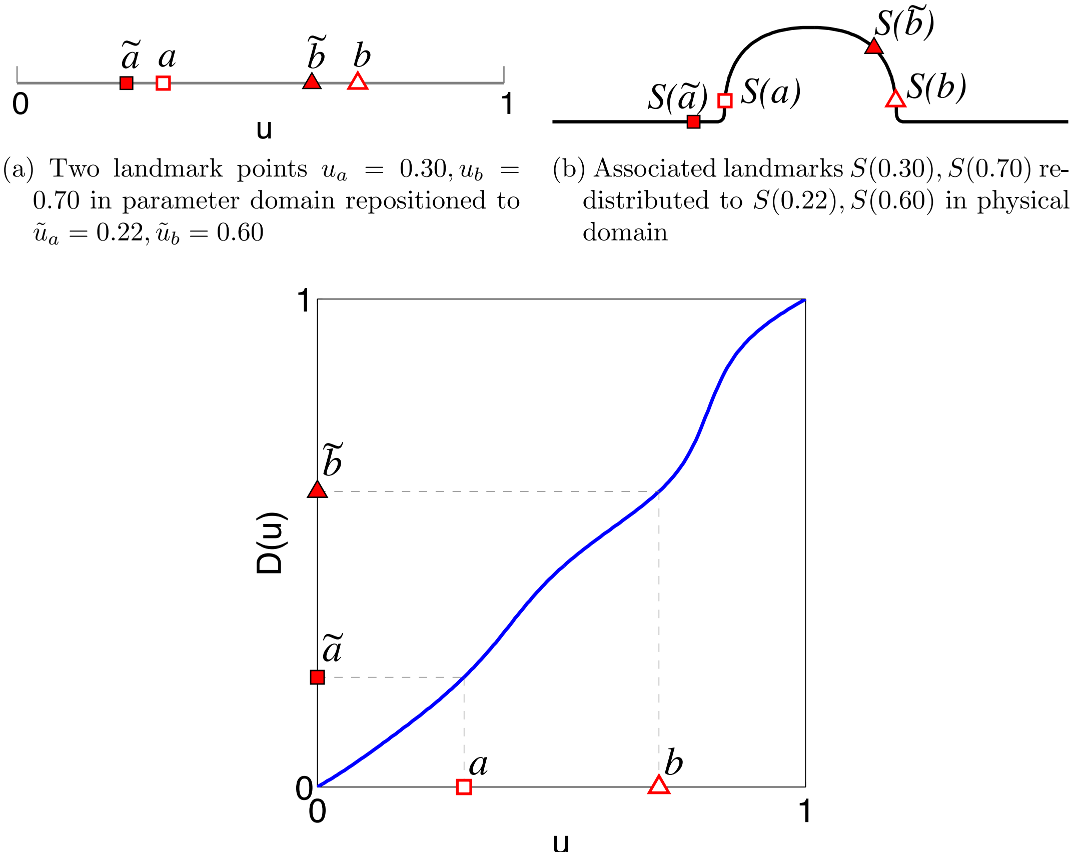
We show below how B-splines can also be used for representing the reparameterization of parametric curves and surfaces.

2.1.1.1 Reparameterization of parametric curves.

Figure 2.1 shows the reparameterization of a 2D curve where points sampled on u have been moved to $D(u)$. Such a reparameterization function $D(u)$ can be represented with the cancatenation of simple mappings. For example, the reparameterization function $D(u)$ shown in Figure 2.1 is represented in Figure 2.2(a) with 4 Cauchy kernels (centered at $\mathbf{c}_1, \mathbf{c}_2, \mathbf{c}_3, \mathbf{c}_4$) which are sequentially superimposed and integrated into $D(u)$ as proposed by [28]; See Appendix I for details regarding Cauchy kernel and its concatenation. .

In this thesis, we propose to directly represent $D(u)$ as a single B-spline function as shown in Figure 2.2(b) where empty circles represent B-spline coefficients $\{b_i\}$. The reparameterization function $D(u)$ for a parametric curve can be represented by a B-spline with n_b coefficients

$$D(u) = \sum_{i=0}^{n_b-1} B_{i,p}(u) b_i, \qquad 0 \leq u \leq 1; \tag{2.2}$$

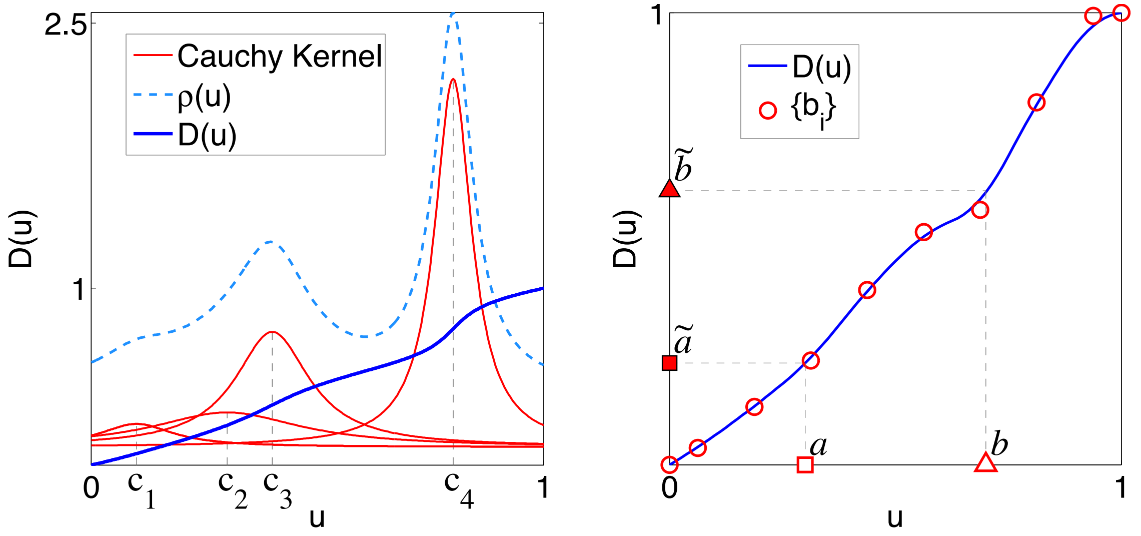


(a) Two landmark points $u_a = 0.30, u_b = 0.70$ in parameter domain repositioned to $\tilde{u}_a = 0.22, \tilde{u}_b = 0.60$

(b) Associated landmarks $S(0.30), S(0.70)$ redistributed to $S(0.22), S(0.60)$ in physical domain

(c) Raparameterization function

Figure 2.1. Raparameterization function for a 2D curve. Hollow landmarks in square and triangle represent sampled points $\mathbf{S}(\mathbf{u})$ based on parameterization $\mathbf{u}$ and solid ones $\mathbf{S}(\mathbf{D}(\mathbf{u}))$ are based on reparameterization function $\mathbf{D}(\mathbf{u})$.



(a) Concatenation of 4 Cauchy kernels centered at c_1, c_2, c_3, c_4

(b) Single B-spline function controlled by B-spline coefficients shown in circles

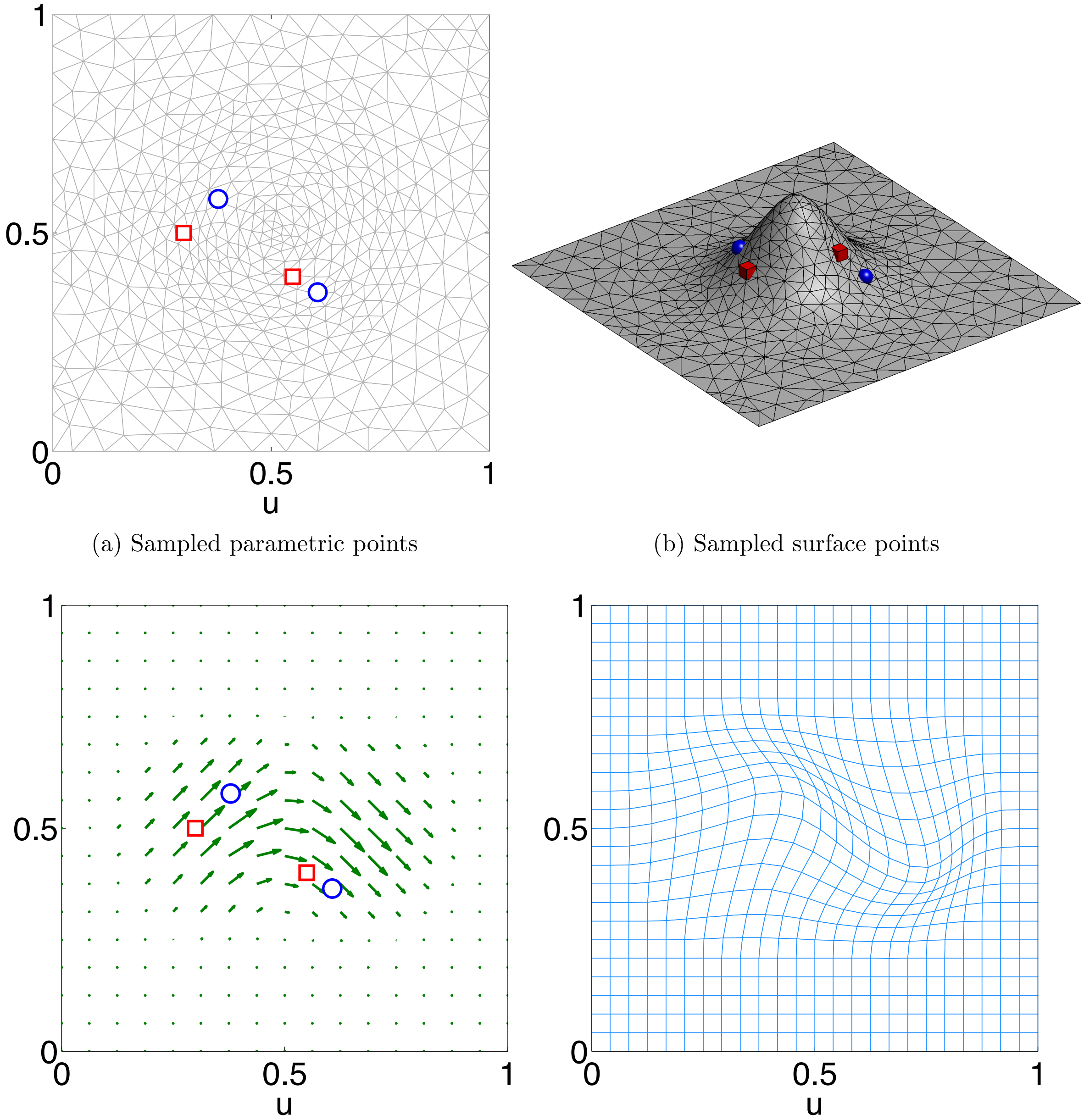
Figure 2.2. Representing reparameterization function for 2D curves: Concatenation of functions vs. single function

where $B_{i,p}$ is the B-spline basis function [32] of degree p associated with the i-th B-spline coefficient b_i, and it is recursively defined on a non-decreasing knot vector $\mathbf{U} = \{\bar{u}_0, \bar{u}_1, \cdots, \bar{u}_{n+p+1}\}$.

We assume that the starting and ending points of all curves are already in correspondence. With this assumption, the boundary of the parameter domain (two ends at $u = 0, 1$) is fixed even with the reparameterization function. Therefore, we use a clamped knot vector (i.e. repeating the first $(p+1)$ and last $(p+1)$ knots) and $b_0 = 0$, $b_n = 1$ for representing the reparameterization so that $D(0) = 0$ and $D(1) = 1$.

2.1.1.2 Reparameterization of parametric surfaces. A reparameterization of a 3D surface is illustrated in Figure 2.3, where squares and circles respectively represent two sampled points before and after the reparameterization. Since a 3D surface $\mathbf{S}(\mathbf{u}) = [x(\mathbf{u}), y(\mathbf{u}), z(\mathbf{u})]$ is mapped to a 2D parametric domain, i.e. $\mathbf{u} = (u, v)$, the reparameterization $\mathbf{D}(\mathbf{u})$ for 3D surfaces $\mathbf{S}(\mathbf{u})$ have two components in u and v

directions $\mathbf{D}(\mathbf{u}) = [D^u(u,v), D^v(u,v)]$. This reparameterization function could also be visualized by a vector field in Figure 2.3(c), and its parametric grid is shown Figure 2.3(d).



(a) Sampled parametric points

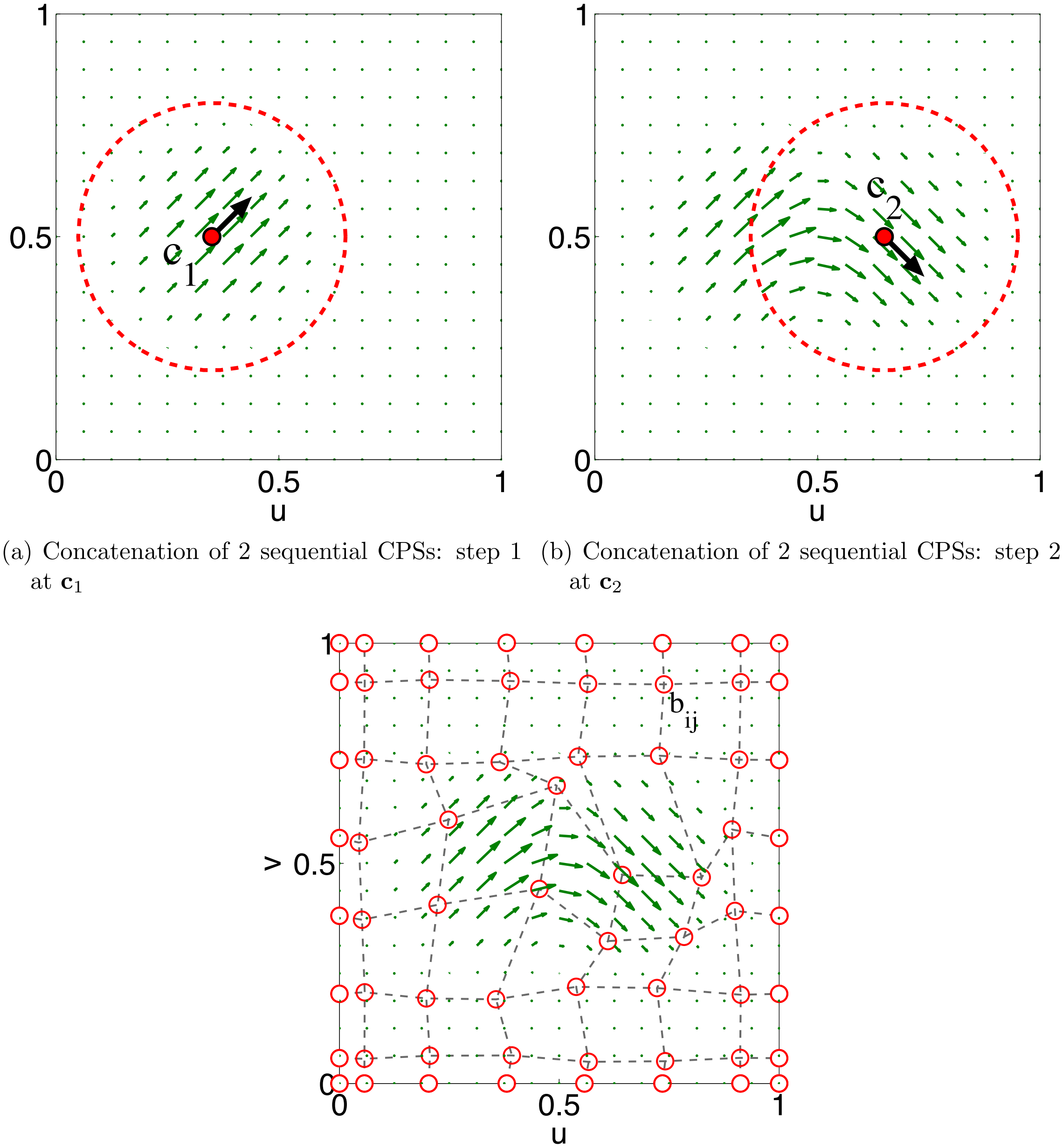
(b) Sampled surface points

(c) Reparameterization function $\mathbf{D}(\mathbf{u}) = (D^u, D^v)$

(d) Reparametrized parametric grid

Figure 2.3. Reparameterization vector field $\mathbf{D}(\mathbf{u})$ and sample redistribution on a 3D surface: cubes and spheres respectively represent points before and after reparameterization.

Such a reparameterization field can be represented by the concatenation of simple mappings. For example, two Clamped Plate Spline (CPS) warps [28] are applied sequentially in the parameter domain shown in Figure 2.4(a) and (b) . These two CPS warps are with centers $\mathbf{c}_1$ and $\mathbf{c}_2$ and red dotted circles as the CPS range. See Appendix II for details on CPS warps.



(a) Concatenation of 2 sequential CPSs: step 1 at $\mathbf{c}_1$

(b) Concatenation of 2 sequential CPSs: step 2 at $\mathbf{c}_2$

(c) Reparameterization by B-splines

Figure 2.4. Representing reparameterization vector field: concatenation of simple mappings vs. single B-spline based mapping.

Instead of using concatenation of simple mappings, we propose the use of single

B-spline functions to directly represent the reparameterization of u and v component of the parametric domain, as shown in Figure 2.4(c) where 8×8 B-spline coefficients b_i (red circles) are used to represent the reparameterization field $\mathbf{D}(\mathbf{u})$. In general, the reparameterization $\mathbf{D}(\mathbf{u}) = [D^u(u,v), D^v(u,v)]$ in the square planar parameter domain is defined as

$$\begin{aligned} D^u(u,v) &= \sum_{i=0}^{n_{b1}-1} \sum_{j=0}^{n_{b2}-1} B_{i,p}(u) B_{j,q}(v) b^u_{i,j}, \\ D^v(u,v) &= \sum_{i=0}^{n_{b1}-1} \sum_{j=0}^{n_{b2}-1} B_{i,p}(u) B_{j,q}(v) b^v_{i,j}, \\ 0 \leq u&, v \leq 1; \end{aligned} \tag{2.3}$$

where $B_{i,p}$ and $B_{j,q}$ are the B-spline basis functions (2.1) of degree p and q associated with the (i,j)-th B-spline coefficient 2-tuple $\mathbf{b}_{i,j} = (b^u_{i,j}, b^v_{i,j})$; the coefficient number along the u- and v-direction are n_{b1} and n_{b2} respectively. They are respectively defined on two sets of non-decreasing knot vector $\mathbf{U}_1$ and $\mathbf{U}_2$.

The four boundaries of all the shapes are assumed to be in correspondence already. With this assumption, the four boundaries of the square parametric domain are fixed during reparameterization, i.e.

$$\begin{aligned} D^u(0,v) = 0, \quad D^u(1,v) = 1, \\ D^v(u,0) = 0, \quad D^v(u,1) = 1. \end{aligned} \tag{2.4}$$

Therefore, the two knot vectors are chosen to be of clamped type and the B-spline coefficients at the four boundaries are either 0 and 1.

2.1.2 Diffeomorphism of reparameterization.

2.1.2.1 Curve case. Setting $\mathrm{d}D(u)/\mathrm{d}u > 0$ in (2.2) gives the bijectivity constraint for diffeomorphic reparameterization of curves for correspondence manipulation. Since $D(u)$ is a degree-p B-spline function, its derivative is a degree-$(p-1)$ B-spline function with $b_{i+1}-b_i$ as B-spline coefficients [34], we thus have the following explicit constraint for ensuring diffeomorphic reparameterization of curves

$$b_i - b_{i+1} < 0, \qquad i = 0, 1, \cdots, n_b - 2. \tag{2.5}$$

2.1.2.2 Surface case. The bijectivity of reparameterization can be guaranteed by the positivity of Jacobian throughout the parameter domain, i.e.

$$J(\mathbf{D}(\mathbf{u})) = \begin{vmatrix} \frac{\partial D^u(u,v)}{\partial u} & \frac{\partial D^u(u,v)}{\partial v} \\ \\ \frac{\partial D^v(u,v)}{\partial u} & \frac{\partial D^v(u,v)}{\partial v} \end{vmatrix} > 0, \quad \forall (u,v) \in [0,1]. \tag{2.6}$$

Assuming there are $n_{b1} \times n_{b2}$ B-spline coefficient tuples $\mathbf{b_{i,j}}$ for representing the reparameterization in (2.3). Due to the boundary constraint (2.4), there are only $(n_{b1}-2) \times (n_{b2}-2)$ interior coefficients $\mathbf{b}_{i,j}$ that can be used to manipulate correspondence for each shape. Consequently, there are $2 \times (n_S - 1) \times (n_{b1} - 2) \times (n_{b2}-2)$ optimization variables for n_S shapes. To ensure the reparameterization $\mathbf{D}(\mathbf{u})$ is diffeomorphic, the bijectivity condition (2.6) that prevents the self-intersection of the parametric field can be cast as constraints on the reparameterization parameters, i.e. interior B-spline coefficient tuples $\mathbf{b}_{i,j}$.

Diffeomorphism via constraints in the B-spline form of Jacobian. We give below a sufficient condition for ensuring the Jacobian field $J(\mathbf{u})$ (2.6) is positive so that the reparameterization $\mathbf{D}(\mathbf{u})$ is diffeomorphic. The scalar Jacobian field $J(\mathbf{u})$ defined in (2.6) for the B-spline based reparameterization (2.3) consists of derivatives of B-splines (piecewise polynomials) and thus remains piecewise polynomials. Therefore, $J(\mathbf{u})$ itself can be cast in the B-spline form as described in [35] [36]. More specifically, the Jacobian $J(\mathbf{u})$ can be expressed as

$$\begin{aligned} J(u,v) &= \det\left[\frac{\partial \mathbf{D}(u,v)}{\partial u}, \frac{\partial \mathbf{D}(u,v)}{\partial v}\right], \\ \frac{\partial \mathbf{D}(u,v)}{\partial u} &= \sum_{i=0}^{n_{b1}-2}\sum_{j=0}^{n_{b2}-1} B_{i,p-1}(u)B_{j,q}(v)\gamma_i\left(\mathbf{b}_{i+1,j}-\mathbf{b}_{i,j}\right), \\ \frac{\partial \mathbf{D}(u,v)}{\partial v} &= \sum_{k=0}^{n_{b1}-1}\sum_{l=0}^{n_{b2}-2} B_{k,p}(u)B_{l,q-1}(v)\eta_l\left(\mathbf{b}_{k,l+1}-\mathbf{b}_{k,l}\right), \end{aligned} \tag{2.7}$$

where: $\gamma_i = \dfrac{p}{\bar{u}_{p+i+1}-\bar{u}_{i+1}}; \eta_l = \dfrac{q}{v_{q+l+1}-v_{l+1}}$. Using the notation $\Delta\mathbf{b}^u_{i,j} = (\mathbf{b}_{i+1,j}-\mathbf{b}_{i,j})$ and $\Delta\mathbf{b}^v_{k,l} = (\mathbf{b}_{k,l+1}-\mathbf{b}_{k,l})$ and noticing that the product of two B-splines is a higher-degree B-spline [37], the Jacobian could be written as the following B-spline form

$$\begin{aligned} J(\mathbf{u}) = J(u,v) = &\sum_{i=0}^{n_{b1}-2}\sum_{j=0}^{n_{b1}-1}\sum_{k=0}^{n_{b1}-1}\sum_{l=0}^{n_{b2}-2} B_{i,p-1}(u)B_{j,q}(v) \\ &B_{k,p}(u)B_{l,q-1}(v)\gamma_s\eta_l \det\left[\Delta\mathbf{b}^u_{i,j}\ \Delta\mathbf{b}^v_{k,l}\right] \\ = &\sum_{s=0}^{2n_{b1}-3}\sum_{t=0}^{2n_{b2}-3} B_{s,2p-1}(u)B_{t,2q-1}(v)J^{BSP}_{s,t}(\{\mathbf{b}\}), \end{aligned} \tag{2.8}$$

where $J^{BSP}_{s,t}$ is the B-spline coefficient of the B-spline form of the Jacobian $J(\mathbf{u})$ for the reparameterization $\mathbf{D}(\mathbf{u})$ in (2.3); each $J^{BSP}_{s,t}$ is a function of $\{\mathbf{b}\}$. The above B-spline form of Jacobian leads to the following sufficient condition for ensuring the reparameterization $\mathbf{D}(\mathbf{u})$ is diffeomorphic. Due to the non-negativeness of B-spline basis functions, $J(\mathbf{u}) > 0$ when every B-spline coefficient in (2.8) is positive, i.e.

$$\begin{gathered} J^{BSP}_{s,t}(\{\mathbf{b}\}) > 0, \\ s = 0, 1, \cdots, n_{J1}-1;\ t = 0, 1, \cdots, n_{J2}-1, \end{gathered} \tag{2.9}$$

where $n_{J1} = 2n_{b1}-2$ and $n_{J2} = 2n_{b2}-2$ are the number of B-spline coefficients of the B-spline form of Jacobian along the u- and v-direction; thus there are total $n_J = n_{J1}n_{J2} = 4(n_{b1}-1)(n_{b2}-1)$ positivity constraints.

The above condition is a sufficient, but not necessary condition. In order to make the condition less conservative, one can extract Bézier patches out of the B-spline representation and obtain more tighter bound of the Jacobian based on Bézier coefficients, as suggested in [35] and [36]. Such Bézier extraction involves the following intermediate steps:

1. Decompose B-spline of control coefficients $\{\mathbf{b}\}$ into Bézier patches (each patch with control coeffcients $\{\widetilde{\mathbf{b}}\}$) by knot insertion algorithms elaborated in [34].
2. Find the Bézier representation of the Jacobian of the Bézier patches as directed in [38].
3. Repose the Jacobian representation of Bézier patches to form the Jacobian B-spline of C^0 inter-patch continuity with control coefficients $\{J^{BEZ}_{s,t}\}$

For each reparameterization $\mathbf{D}(\mathbf{u})$ represented by degree $p \times q$ B-splines with $n_{b1} \times n_{b2}$ coefficient tuples $\mathbf{b}_{i,j}$, there are $(n_{b1} - p) \times (n_{b2} - q)$ Bézier patches with control coeffcients $\{\widetilde{\mathbf{b}}\}$ after decomposition, and there will be $(n_{b1} - p)(2p - 1) + 1$ and $(n_{b2} - q)(2q - 1) + 1$ Jacobian B-spline coefficients along each parametric direction. Therefore, the bijectivity can be ensured by enforcing the following constraints for each shape instance as

$$\begin{gathered} J^{A}_{s,t} = J^{BEZ}_{s,t}(\{\widetilde{\mathbf{b}}\}) > 0 \\ s = 1, \cdots, n_{J1};\ t = 1, \cdots, n_{J2} \end{gathered} \tag{2.10}$$

where $n_{J1} = (n_{b1} - p)(2p - 1) + 1$ and $n_{J2} = (n_{b2} - q)(2q - 1) + 1$ are the number of Bézier coefficients of the Bézier form of Jacobian along the u- and v-direction; thus there are totally $n_J = n_{J1}n_{J2} = [(n_{b1} - p)(2p - 1) + 1][(n_{b2} - q)(2q - 1) + 1]$ positivity constraints.

Jacobian constraints on landmarks. A simple alternative to the above rigorous diffeomorphic conditions is to enforce Jacobian positivity at a finite set of parameter points that correspond to landmarks. The evaluated Jacobian values are all functions of the B-spline coefficients $\{\mathbf{b}\}$ computed by (2.7). The parameter points chosen are usually those associated with the parametric locations of the landmarks. Suppose there are $n_{J1} = n_{P1}$ and $n_{J2} = n_{P2}$ landmarks along the u- and v-direction respectively, and the (s, t)-th parameter point in $\mathbf{u}_{s,t}$. The constraints can be formulated as

$$\begin{gathered} J^B_{s,t} = J(\mathbf{u}_{s,t}, \{\mathbf{b}\}) = \det \begin{bmatrix} \frac{\partial \mathbf{D}(\mathbf{u}_{s,t})}{\partial u} & \frac{\partial \mathbf{D}(\mathbf{u}_{s,t})}{\partial v} \end{bmatrix} > 0 \\ s = 1, \cdots, n_{J1};\ t = 1, \cdots, n_{J2}, \end{gathered} \tag{2.11}$$

where n_{J1} and n_{J2} are the number of sampled Jacobians along the u- and v-direction, in this case equalling the landmark number along each direction; thus there are totally $n_J = n_{J1}n_{J2} = n_{P1}n_{P2}$ positivity constraints. Although the Jacobian between two landmarks with positive Jacobians is not necessarily positive in theory, we find that in all examples reported in this thesis Jacobians are all positive in between with the sufficient landmark resolution chosen. This alternative gives better efficiency than enforcing the constraints in the B-spline form of Jacobian, as shown in Section 2.3.

It should be noted that the proposed B-spline based reparameterization is based on *Free Form Deformation*(FFD) [39]. FFD based techniques have been used extensively, for example, in matching CT/MRI images (image registration) in a multitude of medical imaging applications [40] [41] [42] [43]. The differences in our approach are the following: it is the parametric domain, rather than physical surfaces, that is deformed; our formulation for enforcing diffeomorphic is different; our approach for enforcing the constraints via aggregation shown in next section is also different. It should be noted that splines have been used in landmark matching, e.g. in [44] [45] where the deformation is driven by energy minimizing cost function with various kernels and the diffeomorphism is guaranteed by the flow solutions to ODE. It should also be noted that diffeomorphisms illustrated are akin to fingerprints observed in [46]. There are also alternative for diffeomorphic reparameterization for curves as reported in [47] [48] [49].

2.2 Optimization techniques

2.2.1 Optimization formulation. With the above B-spline representation of reparameterization functions $\mathbf{D}(\mathbf{u})$ and the diffeomorphic conditions, we thus have the following optimization formulation for using B-spline based reparameterization for manipulating shape correspondence:

$$\min_{\mathbf{b}} \quad f(\mathbf{b}) = \sum_{\lambda_i \geq \lambda_{\text{cut}}} \left[1 + \log \frac{\lambda_k(\mathbf{b})}{\lambda_{\text{cut}}} \right] + \sum_{\lambda_k < \lambda_{\text{cut}}} \frac{\lambda_k(\mathbf{b})}{\lambda_{\text{cut}}} \tag{2.12a}$$

$$\text{s.t.} \quad \left[\mathbf{C}^T(\mathbf{b}) \mathbf{C}(\mathbf{b}) \right] \; \mathbf{v}_k(\mathbf{b}) = \lambda_i(\mathbf{b}) \mathbf{v}_k(\mathbf{b}) \tag{2.12b}$$

$$\mathbf{v}_k^T(\mathbf{b}) \mathbf{v}_k(\mathbf{b}) = 1, \quad k = 1, \cdots, n_S \tag{2.12c}$$

$$g_l(\mathbf{b}) < 0, \quad l = 1, \cdots, n_G \tag{2.12d}$$

In this formulation, $\mathbf{b}$ is the set of optimization variables and represents the collection of interior B-spline coefficient tuples $\mathbf{b}$ for $n_s - 1$ shapes, where one shape from the training set is selected as a reference. The objective function $f(\mathbf{b})$ is the simplified description length, which is a function of eigenvalues computed from (2.12b) and (2.12c). The matrix $\mathbf{C}$ is related to the covariance matrix $\mathbf{E}$ by $\mathbf{E} = \mathbf{C}^T\mathbf{C}$ with

$$\mathbf{C} = \frac{\mathbf{X}_c}{\sqrt{(n_S - 1)n_P}}.$$

The constraint (2.12d) represents the diffeomorphic conditions, i.e. (2.5) for curves and (2.10) (2.11) for surfaces, each of which is a function of optimization variables $\mathbf{b}$. It must be pointed out that our proposed method can be applied to any qualifying objective function other than the one stated in (2.12a).

The optimization formulation given in (2.12) leads to a large-scale optimization problem. For SSM of 3D surfaces, there are $2 \times (n_S - 1) \times (n_{b1} - 2) \times (n_{b2} - 2)$ B-spline coefficients as optimization variables with n_G constraints based on (2.10) (2.11), which will be detained in Section 2.2.3.1.

In order to efficiently obtain optimized shape correspondence, we have developed a gradient-based optimization approach. We have derived analytical gradient of the cost function (2.12a) with respect to optimization variables $\mathbf{b}$ with both direct differentiation and the adjoint sensitivity method. We have also developed a technique to approximate the constraints and to aggregate the large number of Jacobian constraints into one constraint in order to speed up the convergence.

Analytical gradient provides an efficient and accurate mean to obtain gradient for optimization. It is especially important in large-scale optimization problems where the finite difference based approach for computing gradient would be inefficient. It turns out the analytical gradient can be derived for all the differentiable intermediate steps since all steps in our formulation, including reparameterization, sampling, alignment, PCA and DL computation are differentiable, the gradient product gives the analytical objective gradient due to the chain rule as following

$$\frac{\mathrm{d}f}{\mathrm{d}b_r} = \sum_i \frac{\partial f}{\partial \lambda_i} \sum_j \frac{\partial \lambda_i}{\partial x_j^A} \sum_k \frac{\partial x_j^A}{\partial x_k^S} \sum_l \frac{\partial x_k^S}{\partial \mathbf{D}(\mathbf{u}_l)} \sum_r \frac{\partial \mathbf{D}(\mathbf{u}_l)}{\partial b_r} \tag{2.13}$$

where $\mathbf{D}(\mathbf{u}_l)$ is the l-th reparameterized landmark point in the parameter domain, x_k^S is the k-th landmark in the physical domain and x_j^A is the j-th aligned landmark. Among them the $\frac{\partial f}{\partial \lambda_i}$, $\frac{\partial \lambda_i}{\partial x_j^A}$ and $\frac{\partial x_j^A}{\partial x_k^S}$ are inherently differentiable, and $\frac{\partial \mathbf{D}(\mathbf{u}_l)}{\partial b_r}$ is also differentiable as long as a differentiable reparameterization technique such as (2.2) and (2.3) is used. The analytical gradient $\frac{\partial x_k^S}{\partial \mathbf{D}(\mathbf{u}_l)}$ requires the differentiability of the geometric representation of training set shapes. In this thesis, we use the quadratic B-spline of C^1 smoothness to represent the shapes in the training set. Therefore, full analytical gradients can be derived.

2.2.2 Analytical gradient of objective function.

The total sensitivity from

(2.12a) is just

$$\frac{\mathrm{d}\, f(\mathbf{b})}{\mathrm{d}\, b_r} = \sum_i \frac{\partial f[\mathbf{b}, \boldsymbol{\lambda}(\mathbf{b})]}{\partial \lambda_i} \frac{\partial \lambda_i(\mathbf{b})}{\partial b_r} \tag{2.14}$$

where $r = 1, \cdots, n_D$; the optimization variable number $n_D = (n_b - 2)(n_S - 1)$ for curves and $n_D = 2(n_{b1} - 2)(n_{b2} - 2)(n_S - 1)$ for surfaces.

Under the state equation formulation in (2.12), we can obtain the objective sensitivity in (2.14) with either the direct method or the adjoint method, giving rise to the *direct* sensitivity and *adjoint* sensitivity. The adjoint sensitivity is significantly faster than the direct sensitivity. We provide the derivations for both to highlight the characteristics of the adjoint method.

2.2.2.1 Direct sensitivity. The direct sensitivity computes the gradient of the cost function by directly differentiating the cost function (2.12a). That is,

$$\frac{\partial\, f(\mathbf{b})}{\partial\, b_r} = \sum_{m=1}^{n_S} \frac{\partial f}{\partial \lambda_m} \frac{\partial \lambda_m}{\partial b_r}. \tag{2.15}$$

The term $\partial \lambda_m / \partial b_r$ $(m = 1, \cdots, n_S)$ is obtained by differentiating equations (2.12b) and (2.12c) with respect b_r, which leads to the following resulting linear system

$$\begin{bmatrix} 0 & \mathbf{v}_m^T \\ \mathbf{v}_m & \lambda_m \mathbf{I}_{n_S} - \mathbf{C}^T\mathbf{C} \end{bmatrix} \begin{bmatrix} \frac{\partial \lambda_m}{\partial b_r} \\ \frac{\partial \mathbf{v}_m}{\partial b_r} \end{bmatrix} = \begin{bmatrix} 0 \\ \frac{\partial (\mathbf{C}^T\mathbf{C})}{\partial b_r} \mathbf{v}_m \end{bmatrix}. \tag{2.16}$$

The linear equation system can be solved to obtain $\partial \lambda_m / \partial b_r$ and $\partial \mathbf{v}_m / \partial b_r$. Plugging $\partial \lambda_m / \partial b_r$ into (2.15) yields the desired gradient.

2.2.2.2 Adjoint sensitivity. In order to avoid the direct computation of $\partial \lambda_m / \partial b_r$ and $\partial \mathbf{v}_m / \partial b_r$, we introduce a Lagrangian quantity by augmenting the objective function (2.12a) with $2n_S$ sets of constraints (2.12b) and (2.12c) as

$$\mathcal{L} = f(\boldsymbol{\lambda}) + \sum_{m=1}^{n_S} \boldsymbol{\mu}_m^T \left(\mathbf{C}^T\mathbf{C}\, \mathbf{v}_m - \lambda_m \mathbf{v}_m \right) + \sum_{m=1}^{n_S} \nu_m (\mathbf{v}_m^T \mathbf{v}_m - 1) \tag{2.17}$$

where adjoint variables $\boldsymbol{\mu}_m$ and ν_m $(m = 1, \cdots, n_S)$ are the Lagrange multipliers.

Differentiation of (2.17) w.r.t b_r gives

$$\begin{aligned}\frac{\partial f}{\partial b_r} = \frac{\partial \mathcal{L}}{\partial b_r} = &\sum_{m=1}^{n_S} \frac{\partial f}{\partial \lambda_m}\frac{\partial \lambda_m}{\partial b_r} \\ &+ \sum_{m=1}^{n_S} \boldsymbol{\mu}_m^T \left(\frac{\partial (\mathbf{C}^T\mathbf{C})}{\partial b_r} - \frac{\partial \lambda_m}{\partial b_r}\mathbf{I}_{n_S} \right) \\ &+ \sum_{m=1}^{n_S} \boldsymbol{\mu}_m^T \left(\mathbf{C}^T\mathbf{C} \ - \lambda_m \mathbf{I}_{n_S} \right) \frac{\partial \mathbf{v}_m}{\partial b_r} + \sum_{m=1}^{n_S} 2\nu_m \mathbf{v}_m^T \frac{\partial \mathbf{v}_m}{\partial b_r},\end{aligned}$$

which could be simplified to

$$\begin{aligned}\frac{\partial f}{\partial b_r} = &\sum_{m=1}^{n_S} \boldsymbol{\mu}_m^T \frac{\partial (\mathbf{C}^T\mathbf{C})}{\partial b_r}\mathbf{v}_m + \sum_{m=1}^{n_S} \left(\frac{\partial f}{\partial \lambda_m} - \boldsymbol{\mu}_m^T \mathbf{v}_m \right) \frac{\partial \lambda_m}{\partial b_r} \\ &+ \sum_{m=1}^{n_S} \left[\boldsymbol{\mu}_m^T (\mathbf{C}^T\mathbf{C} - \lambda_m \mathbf{I}_{n_S}) + 2\nu_m \mathbf{v}_m^T \right] \frac{\partial \mathbf{v}_m}{\partial b_r}.\end{aligned} \tag{2.18}$$

The key idea of the adjoint method is to circumvent the direct computation of the sensitivity of the state variables λ with respect to optimization variables $\boldsymbol{b}$. A more general discussion on the adjoint method is available in [50] [51] [52]. Specifically in this situation, in order to bypass the direct calculation of $\frac{\partial \lambda_m}{\partial b_r}$ and $\frac{\partial \mathbf{v}_m}{\partial b_r}$ in (2.18), their coefficients in (2.18) are set to zero. This is possible since the arbitrariness of adjoint variables $\boldsymbol{\mu}_m$ and ν_m. This therefore leads to the linear adjoint equations below for solving these adjoint variables

$$\begin{cases} \dfrac{\partial f}{\partial \lambda_m} - \boldsymbol{\mu}_m^T \mathbf{v}_m = 0, \\ \boldsymbol{\mu}_m^T (\mathbf{C}^T\mathbf{C} - \lambda_m \mathbf{I}_{n_S}) + 2\nu_m \mathbf{v}_m^T = \mathbf{0}. \end{cases} \tag{2.19}$$

The above equation can be rearranged into the following linear system

$$\begin{bmatrix} \mathbf{v}_m^T & 0 \\ \mathbf{C}^T\mathbf{C} - \lambda_m \mathbf{I}_{n_S} & 2\mathbf{v}_m \end{bmatrix} \begin{bmatrix} \boldsymbol{\mu}_m \\ \nu_m \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial \lambda_m} \\ \mathbf{0} \end{bmatrix} \tag{2.20}$$

from which the adjoint variables $\boldsymbol{\mu}_m$ and ν_m can be solved. The sensitivity in (2.18) can then be obtained through the following simplified expression

$$\frac{\partial f}{\partial b_r} = \sum_{m=1}^{n_S} \boldsymbol{\mu}_m^T \frac{\partial (\mathbf{C}^T\mathbf{C})}{\partial b_r}\mathbf{v}_m. \tag{2.21}$$

It is worth noting that, with the direct method, the linear system (2.16) is solved for each optimization variable b_r, r $=1$ to n_D, and every eigenvalue λ_m. On the other hand, with the adjoint method, the linear system (2.20) is solved just once for every eigenvalue λ_m.

For the purpose of clearly demonstrating the properties and advantages of analytical gradient formulas presented here, a few computational experiments are conducted on concrete examples as will be discussed in Section 2.3.3.

2.2.3 Inequality constraints for optimization.

2.2.3.1 Constraints aggregation. Based on the description in Section 2.1.2.2, we consider the two options: a) Jacobian B-spline (via Bézier implementaion) (2.10); and b) sampled Jacobian (2.11). In either situation, the large quantity of original n_J constraints is rather undesirable for the optimization with already many optimization variables. For the purpose of reducing the number of constraints while not sacrificing the differentiability of constraint functions, the constraint aggregation [53] [54] technique is employed here. The Kreisselmeier-Steinhauser (KS) function [55] is used here and we choose to aggregate all the n_J Jacaobian constraints for each shape into one single constraint to be applied as the optimization constraints in (2.12d), yielding totally $n_G = n_S - 1$ constraints for $n_S - 1$ shapes as

$$g_l(\mathbf{b}) = \frac{1}{K} \ln \left[\sum_{z=1}^{n_J} e^{-K J_z(\{\mathbf{b}\}_l)} \right] < 0, \quad l \in 1, .., n_S;\ l \neq i_{\mathrm{R}} \tag{2.22}$$

where J_z is original Jacobian positivity, being either $J^A_{s,t}$ in (2.10) for the Jacobian B-spline constraint or $J^B_{s,t}$ in (2.11) for the direct sampled Jacobian constraint; the linearly ordered index is obtained by $z = s + (t-1)n_{J1}$. i_{R} is the reference shape index whose parameter domain is not reparameterized and landmarks are fixed. $\{\mathbf{b}\}_l$ are just the reparameterization B-spline control coefficients for the l-th shape. Parameter K is a control parameter here chosen to be $K = 15$.

It could be shown that this new constraint is more conservative than the original constraints. We have found that the two options of Jacobian positivity constraints differ only slightly as long as the landmark resolution of the direct sampled Jacobian constraint option is sufficiently large.

2.2.3.2 Analytical gradient of constraints. For a gradient based optimization approach, both the gradient of the cost function and the constraints with respect to optimization variables are needed. The gradients of the objective function have been discussed in great details in the previous section. The analytical gradients of the inequality constraints based on (2.22) can also be derived. The gradient of l-th inequality constraint w.r.t the r-th optimization variable b_r is

$$\frac{\partial g_l}{\partial b_r} = -\frac{1}{\sum_{z=1}^{n_J} e^{-KJ_z(\{\mathbf{b}\}_l)}} \sum_{z=1}^{n_J} e^{-KJ_z(\{\mathbf{b}\}_l)} \frac{\partial J_z(\{\mathbf{b}\}_l)}{\partial b_r} \tag{2.23}$$

where the Jacobian gradient $\frac{\partial J_z(\{\mathbf{b}\}_l)}{\partial b_r}$ when $J_{s,t}^A$ and $J_{s,t}^B$ can be evaluated easily from (2.10) and (2.11).

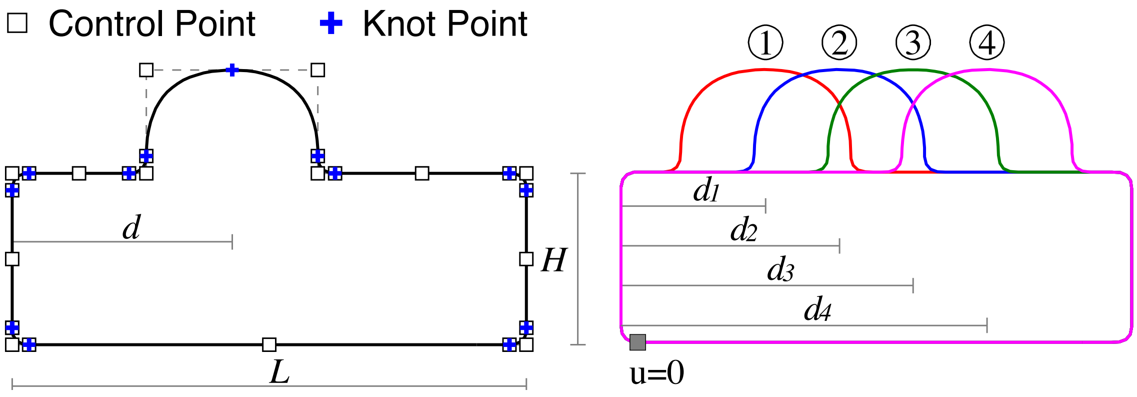
2.3 Experimental results

In this section, we present the numerical results of the proposed correspondence optimization approach. Experimental results on both synthetic and real 2D and 3D data sets are described. The use of synthetic data is to check the correctness of the results since the shape variation pattern is known. The two synthetic data sets are the “box-bump” (2D) and the “plane-bump” (3D). The use of three real data sets, the “hand” (2D), “distal femur” (3D) and “aortic” models, is to show the practicality of the proposed approach.

The optimizer is *Sequential Quadratic Programming* routine in the *MATLAB* optimization toolbox. The stopping criterion for the optimizer is chosen as the relative change of objective function, i.e. $\frac{f^{(k)}-f^{(k-1)}}{f^{(0)}} < 10^{-6}$; it is used both in the concatenation of simple mappings and direct reparameterization schemes. The allowed number of iterations in concatenation of simple mappings based correspondence optimization is set to be $N_{\text{OPT}} = 10^6$ in 2D case and $N_{\text{OPT}} = 10^5$ in 3D case. Other optimization parameters specific to each problem will be explained later.

The points $\mathbf{S}(\mathbf{u}_i^k)$ used for computing the variation in (1.4) are uniformly sampled in the parametric domain. The selection of reference shape $\mathbf{S}_R$ has no noticeable influence on the correspondence result as demonstrated in results below. In order to obtain a unique solution, one *reference* shape instance is chosen out the n_S shapes, so that the optimization's goal is to find $n_S - 1$ reparameterization functions for the remaining shapes that minimize the objective function.

2.3.1 2D synthetic data: box-bump.

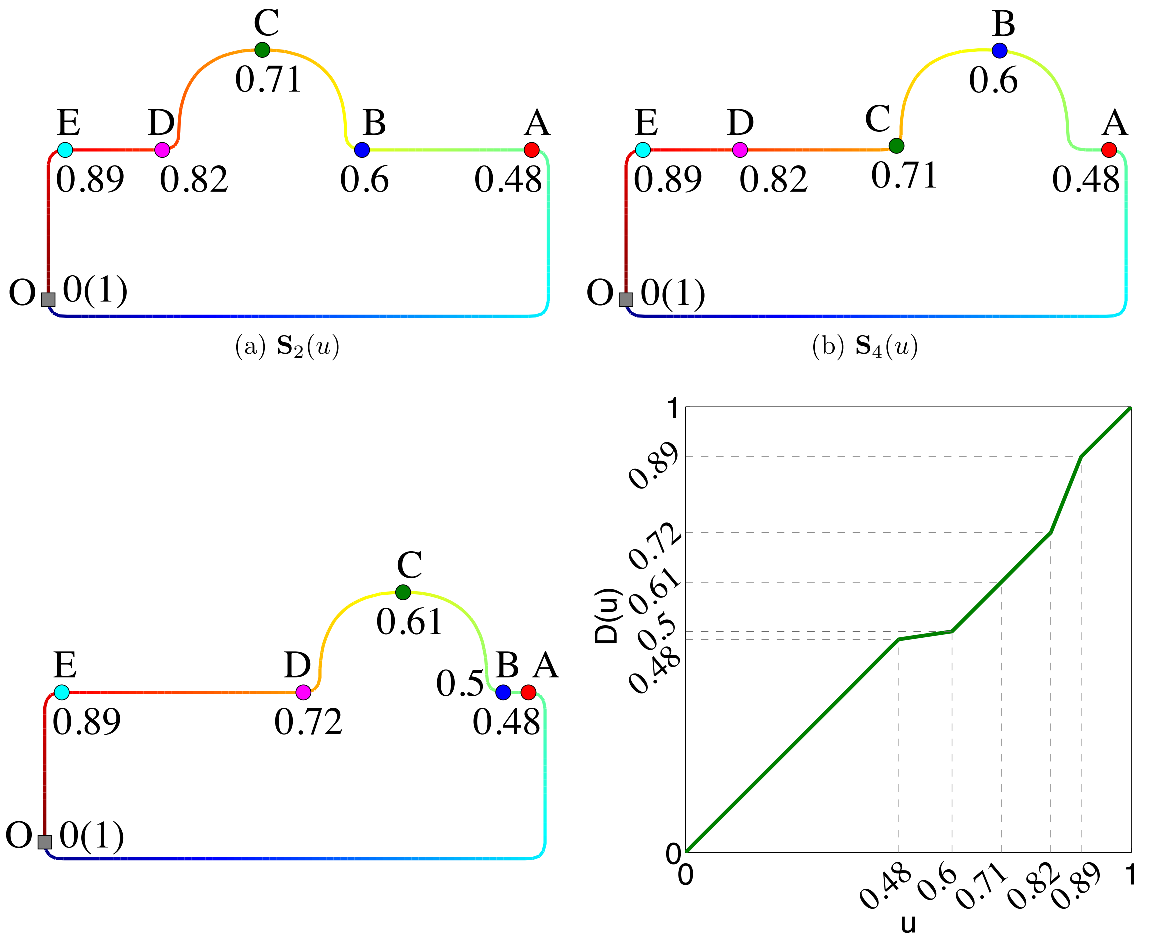


(a) B-spline representation of box-bump geometry

(b) 4 box-bumps superimposed

Figure 2.5. Four B-spline represented box-bumps.

The box-bump example has been extensively used as a benchmark example for testing statistical shape modeling algorithms [28] [24] [26] [22]. The only difference here is that the training set is now represented by B-splines instead of polylines. Key geometry dimensions are plotted in Figure 2.5(a), where the box length and height are both fixed at $L = 60$ and $H = 20$. Different shape instances may have different horizontal distance d between the bump center and the left side. Four shape instances are generated, and their key dimensions are $d_1 = 15, d_2 = 25, d_3 = 35, d_4 = 45$ as shown in Figure 2.5(b).



(a) $\mathbf{S}_2(u)$

(b) $\mathbf{S}_4(u)$

(c) $\mathbf{S}_4(D_4^*(u))$

(d) Desired $D_4^*(u)$ for Shape 4, by comparing (b) and (c)

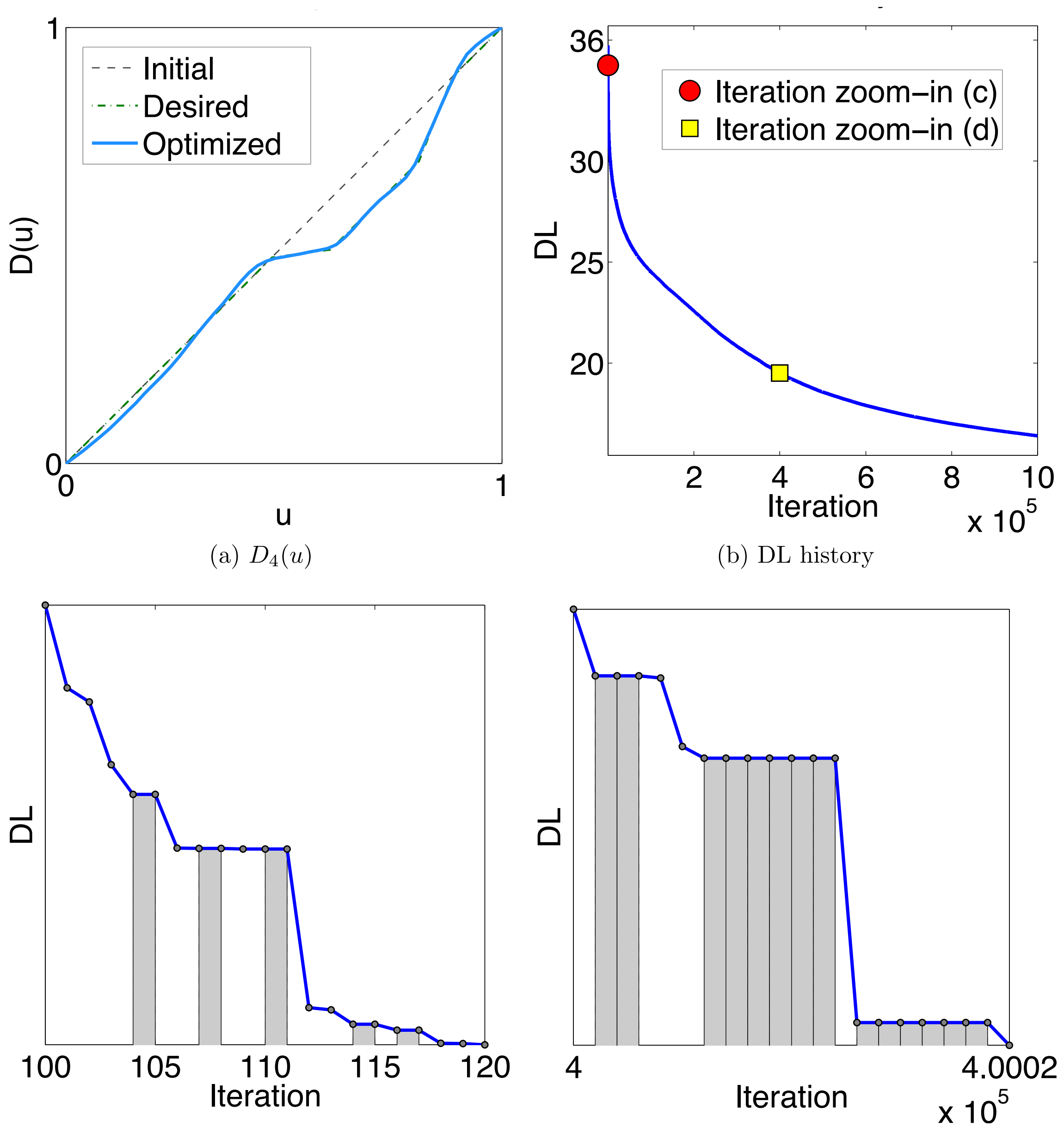
Figure 2.6. Desired reparameterization function for the 4th-shape $D_4^*(u)$ with shape 2 as the reference.

2.3.1.1 Desired reparameterization function. The use of synthetic data makes it possible to obtain the desired reparameterization $D^*(u)$ manually and compare it with the optimized reparameterization to see the quality of correspondence optimization. Without loss of generality, we choose shape 2 as the reference shape and the sample points $\mathbf{S}_2(u)$ on shape 2 at parameters $u = \{0.48, 0.6, 0.71, 0.82, 0.89\}$ are shown in Figure 2.6(a). At the same parameters, the sampled points on $\mathbf{S}_4(u)$ are shown in Figure 2.6(b), with apparent poor correspondence with the points on shape 2. Since the B-spline represented curves are parameterized with segment length measured from the starting point $u = 0$ shown in gray square in Figure 2.5(b). One can obtain the desired reparameterization function $D_4^*(u) = \{0.48, 0.6, 0.71, 0.82, 0.89\}$ for the shape 4 by mapping corner points in shape 4 with those in shape 2. The sampled points $S_4[D_4^*(u)]$ and D_4^* are respectively shown in Figure 2.6(c) and Figure 2.6(d). D_1^* and D_3^* can be formed similarly but not shown here.

2.3.1.2 Reparameterization by concatenation of Cauchy kernels.

Before optimization, 51 landmarks are uniformly sampled in the parameter domain for all the four shape instances. Shape 2 is chosen as reference. The concatenation of Cauchy kernels scheme is able to obtain reparameterization functions very close to desired reparameterization functions $\{D_k^*(u)\}(k = 1, 3, 4)$. After optimization w.r.t the magnitudes of randomly generated Cauchy kernels $N_{\text{OPT}} = 10^6$ times, the reparameterization function of Shape 4 $D_4(u)$ is very close to the desired $D_4^*(u)$ as shown in Figure 2.7(a). Figure 2.7(b) shows the convergence history of the description length where the DL drops considerably from 35.2 to 15.1 after 10^6 iterations. However, this optimization approach is extremely slow since it costs 7.6 hours even for this small population of shapes. In fact, 73% of the one million iterations does not even reduce DL due to two reasons. First, due to the limited optimization space afforded in each Cauchy kernel, every iteration (optimization with one Cauchy kernel) only leads to small reduction of the cost function. Second, after the optimization with one Cauchy kernel, there is no gradient information guiding the selection of subsequent Cauchy kernels to reduce the description length. As such, there may be no reduction in DL between consecutive Cauchy kernels based optimization runs. Such objective function “plateau” during the convergence process is plotted in two zoomed in locations with 20 iterations shown in Figure 2.7(c)(d). It can be seen that during many adjacent iterations, the DL does not decrease.

2.3.1.3 Reparameterization via B-splines.



(a) $D_4(u)$

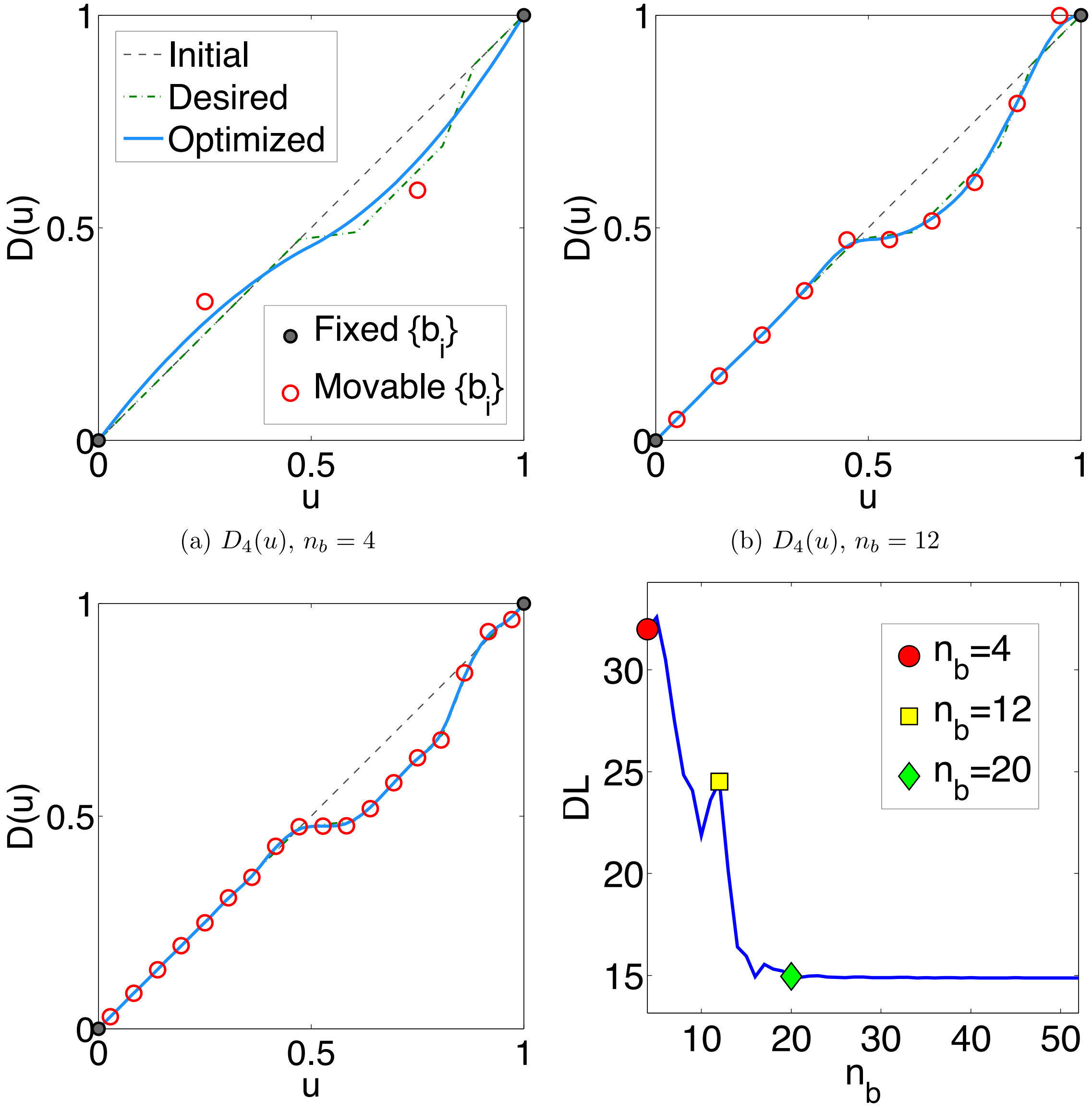
(b) DL history

(c) The 20 iterations: $100 \sim 120$-th: 5 iterations with no reduction in DL

(d) The 20 iterations: $400,000 \sim 400,020$-th: 14 iterations with no reduction in DL

Figure 2.7. Optimized reparameterization with concatenation of Cauchy kernels and DL history with zoom-in; gray bars highlight iterations with no reduction in DL

Optimized reparameterization function. Figure 2.8 shows the proposed



(a) $D_4(u)$, $n_b = 4$

(b) $D_4(u)$, $n_b = 12$

(c) $D_4(u)$, $n_b = 20$

(d) DL convergence w.r.t n_b

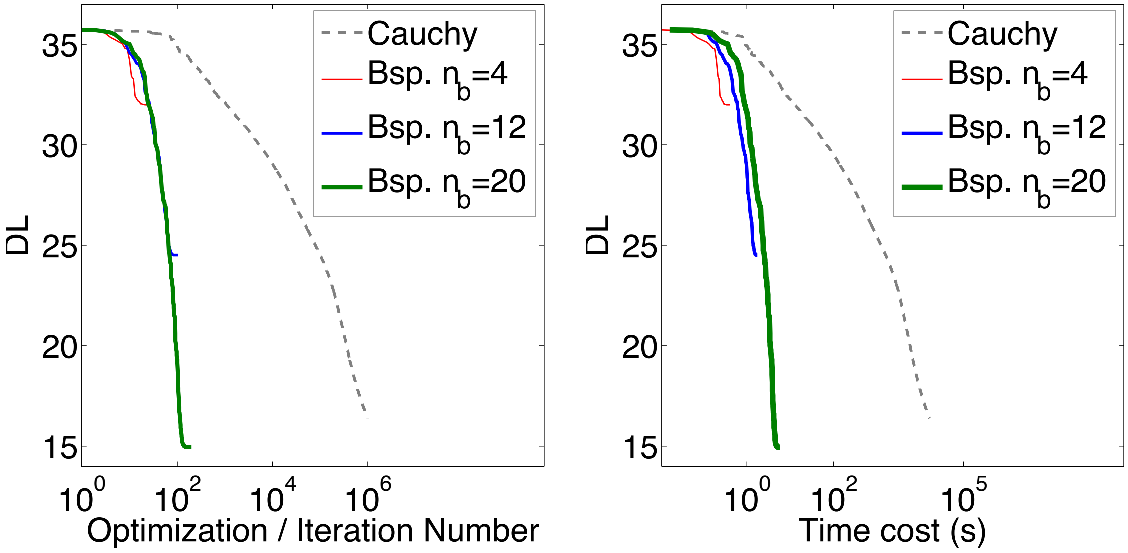
Figure 2.8. Optimized reparameterization function $D(u)$ at different different reparameterization B-spline resolution n_b and convergence of optimized DL w.r.t n_b

direct diffeomorphic reparameterization can efficiently find the optimized $D(u)$ that are very close to the desired ones $D^*(u)$. Figure 2.8(a) shows the optimized $D_4(u)$ with 4 B-spline coefficients to represent each $D(u)$ where only the two internal B-spline coefficients are allowed to vary to improve the correspondence. It takes 24 iterations to converge and the objective function drops from 35.7 to 32.0 and the optimized reparameterization functions are also plotted in Figure 2.8(a). When the number of B-spline coefficients for each $D(u)$ increases to 12 and 20, the optimized reparameterization functions become closer to the desired ones with the optimized function for the 4th shape $D_4(u)$ as shown in Figure 2.8(b) and (c) respectively. Figure 2.8(d) shows that with the increase of B-spline coefficients representing the reparameterization function, the resulting description length becomes smaller and eventually converges, where the three states of $n_b = 4, 12, 20$ in colored markers associated with Figure 2.8(a)(b)(c) respectively are highlighted on the DL convergence curve.

Time cost comparison. Figure 2.9 shows the time cost comparison between the concatenation of Cauchy kernels and the B-spline based direct reparameterization approaches. It shows in order to reduce the DL objective function to the same level around 15, the direct reparameterization takes about 100 iterations and 10 seconds, which is four orders of magnitude faster than the concatenation of Cauchy kernel based approach (10^6 iterations in 10^5 seconds).

Correspondence improvement with optimization. Figure 2.10 demonstrates the significant correspondence improvement as observed on Shape 4. Before optimization, the five points on Shape 4 shown in Figure 2.10(b) corresponding to the five feature points fixed on the reference Shape 2 in Figure 2.10(a) indicate a poor initial correspondence (Point B signifies right bump corner on Shape 2, but it lies at bump top on Shape 4). After optimization with with $n_b = 22$ reparameterization B-spline control coefficients, the five corresponding feature points on Shape 4 are brought toward the expected feature locations as shown in Figure 2.10(c).

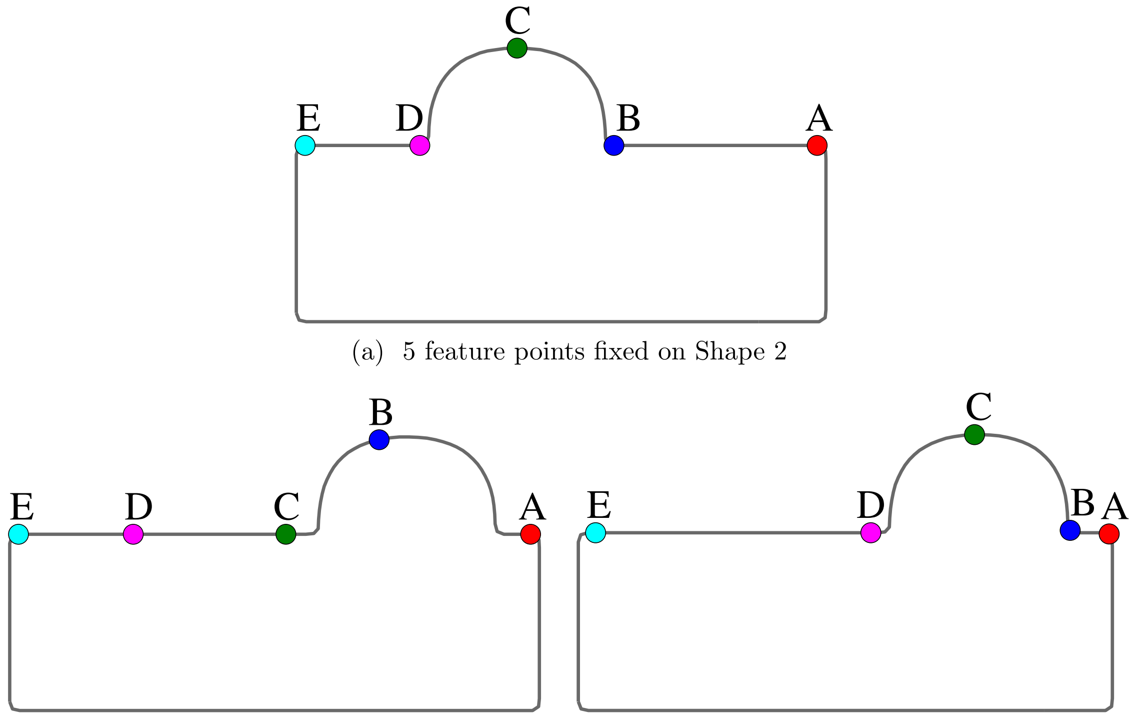
SSM improvement after optimization. As a direct result of the correspondence improvement, the statistical model is optimized where the 1st mode is able to faithfully capture the horizontal sliding pattern as displayed in Figure 2.11(d)(e)(f)



(a) DL w.r.t optimization runs (Cauchy kernels) and iteration number (B-splines)

(b) DL w.r.t time cost

Figure 2.9. Time cost comparison: concatenation of Cauchy kernels vs. B-spline based direct reparameterization



(a) 5 feature points fixed on Shape 2

(b) Badly corresponded feature points on Shape 4 before optimization

(c) Well corresponded feature points on Shape 4 after optimization

Figure 2.10. Correspondence before and after optimization with $n_b = 22$ B-spline coefficients in reparameterization B-spline.

while initial statistical model fails to do so as shown in Figure 2.11(a)(b)(c). From a quantitative perspective, the initial SSM has variation $\lambda_1 = 5.78$ accounting for only 73% of the total shape variation, whereas the the optimized SSM has variation $\lambda_1 = 34.18$ accounting for all 100% of the total shape variation.

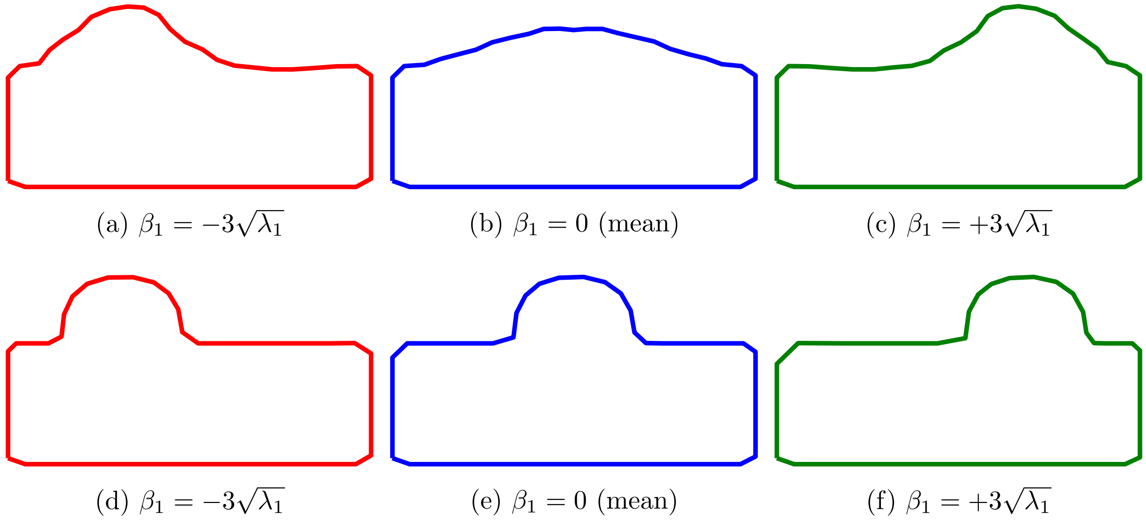

(a) $\beta_1 = -3\sqrt{\lambda_1}$
(b) $\beta_1 = 0$ (mean)
(c) $\beta_1 = +3\sqrt{\lambda_1}$
(d) $\beta_1 = -3\sqrt{\lambda_1}$
(e) $\beta_1 = 0$ (mean)
(f) $\beta_1 = +3\sqrt{\lambda_1}$

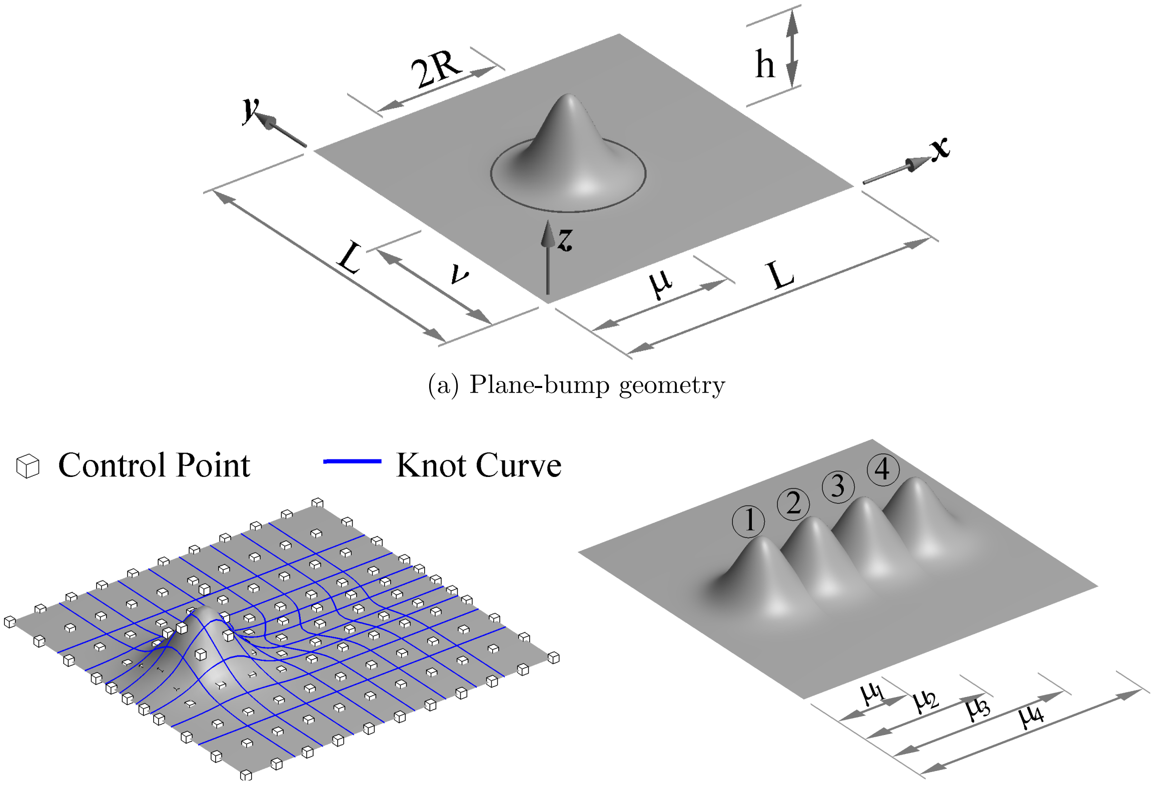
Figure 2.11. SSM's 1st mode before optimization in (a)(b)(c) with variation $\lambda_1 = 5.78$ (73%), and after optimization in (d)(e)(f) with variation $\lambda_1 = 34.18$ (100%)

2.3.2 3D synthetic data: Plane-bump.

Besides the above box-bump curve example, we also test the correspondence with a synthetic plane-bump surface example: the surface is a bivariate Gaussian distribution function with surface point (x, y, z) defined by

$$z = he^{-\frac{(x-\mu)^2}{2\sigma^2}} e^{-\frac{(y-\nu)^2}{2\phi^2}} \tag{2.24}$$

where (μ, ν) is the mean and $\sigma = \phi$ are the standard deviation along x and y direction; h is the peak height. Restricting the distribution function to a square domain $[0, L]\times[0, L]$ that includes the mean point (μ, ν) gives the plane-bump geometry shown in Figure 2.12(a). Such a plane-bump shape can be approximated by a B-spline surface and Figure 2.12(b) show the shape represented by a bi-quadratic B-spline with 11×11 control points that fits the underlying Gaussian distribution surface. If $\nu = \frac{L}{2}$ and R are both locked and only the horizontal position along the x-axis is allowed to move, then a training set of several synthetic plane-bump instances each of which is represented by B-splines are generated. The true shape variation pattern is obviously the translational motion along the x-axis. Four shapes are generated as seen in Figure 2.12(c), and their mean positions $\mu_1, \mu_2, \mu_3, \mu_4$ are uniformly spaced. Each of the four plane-bump B-spline instances inherently implies a parameterization. During the generation of parameterization, attention is needed to introduce as little parameterization distortion as possible so as to greatly reduce the chance of under-sampling in later procedure. Figure 2.12(b) shows the parameterization with fairly low distortion as indicated by its knot curves for Shape 1; the other three instances are similarly parameterized.



(a) Plane-bump geometry

(b) B-spline represented plane-bump

(c) 4 instances superimposed

Figure 2.12. Plane-bump geometry and training set of 4 B-spline surfaces

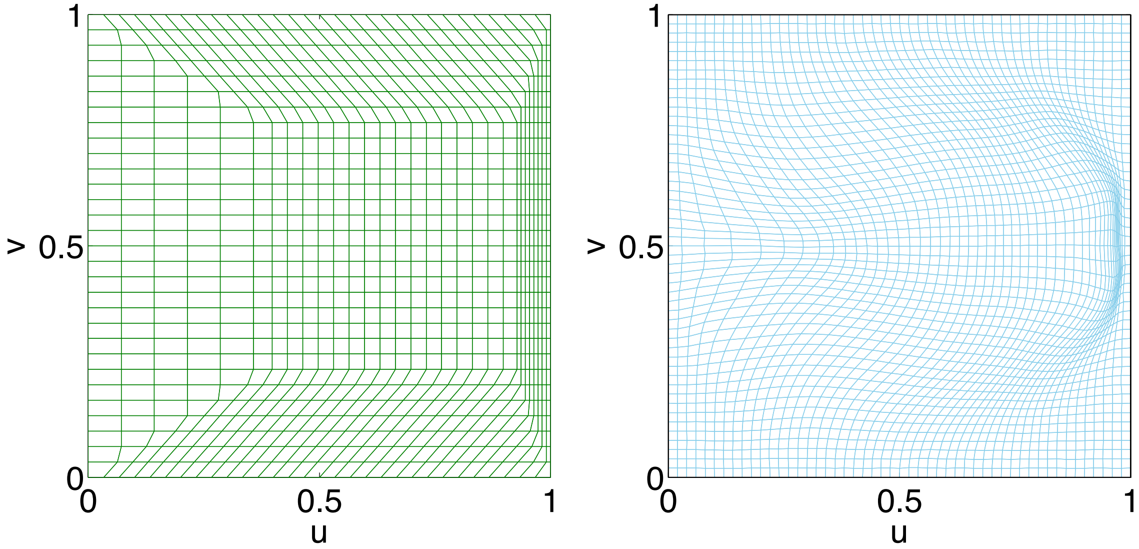
2.3.2.1 Desired reparameterization function. Due to the different bump locations along the x-axis, the bumps for four shapes correspond to different areas in the parameter domain. This difference becomes very clear when we pick the same point in the parameter domain and compare their associated points in the physical domain for all the instances. Without loss of generality, we pick five parameter points: $\mathbf{u}_A = (0.45, 0.5), \mathbf{u}_B = (0.17, 0.5), \mathbf{u}_C = (0.73, 0.5), \mathbf{u}_D = (0.45, 0.22), \mathbf{u}_E = (0.45, 0.78)$, associated with the five feature points on the bump of Shape 2 as shown in Figure 2.16(a). Sampled at this common parameter point set, the physical points on the four shapes are different; the five points on Shape 4 are taken as an example in Figure 2.16(b). The five feature points on Shape 2 denote the bump peak and four bump base quadrant points; they cover a square region in the parameter domain.

In the subsequent correspondence optimization, Shape 2 is chosen as the reference shape whose landmarks are kept fixed. The desired reparameterization functions for the other three shapes should be one that roughly translate the square region of Shape 2 to those on other shapes. Figure 2.13(a) gives a rough visualization of the parametric grid under the desired reparameterization function for Shape 4, i.e. the translational motion of square region from the left side (near Shape 2's bump area) toward the right side (near Shape 4's bump area).

2.3.2.2 Reparameterization by concatenation of Clamped Plate Splines.

It's possible to obtain the optimal correspondence through two reparameterization methods: concatenation of CPSs [28] and B-splines. There are 51×51 landmark points ($n_P = 2601$) uniformly sampled in each shape; Shape 2 is chosen as the reference shape. After $N_{\mathrm{OPT}} = 10^5$ iterations of CPS warps, the deformed landmark grid (originally a regular 51×51 grid) under the optimized reparameterization function $\mathbf{D}_4(\mathbf{u})$ in Figure 2.13(b) approaches the desired trend shifting rightward along the x-axis very similar to Figure 2.13(a). The reparameterization functions for Shape 2 and 3 are not shown but also exhibit similar trends.

2.3.2.3 Reparameterization via B-splines.

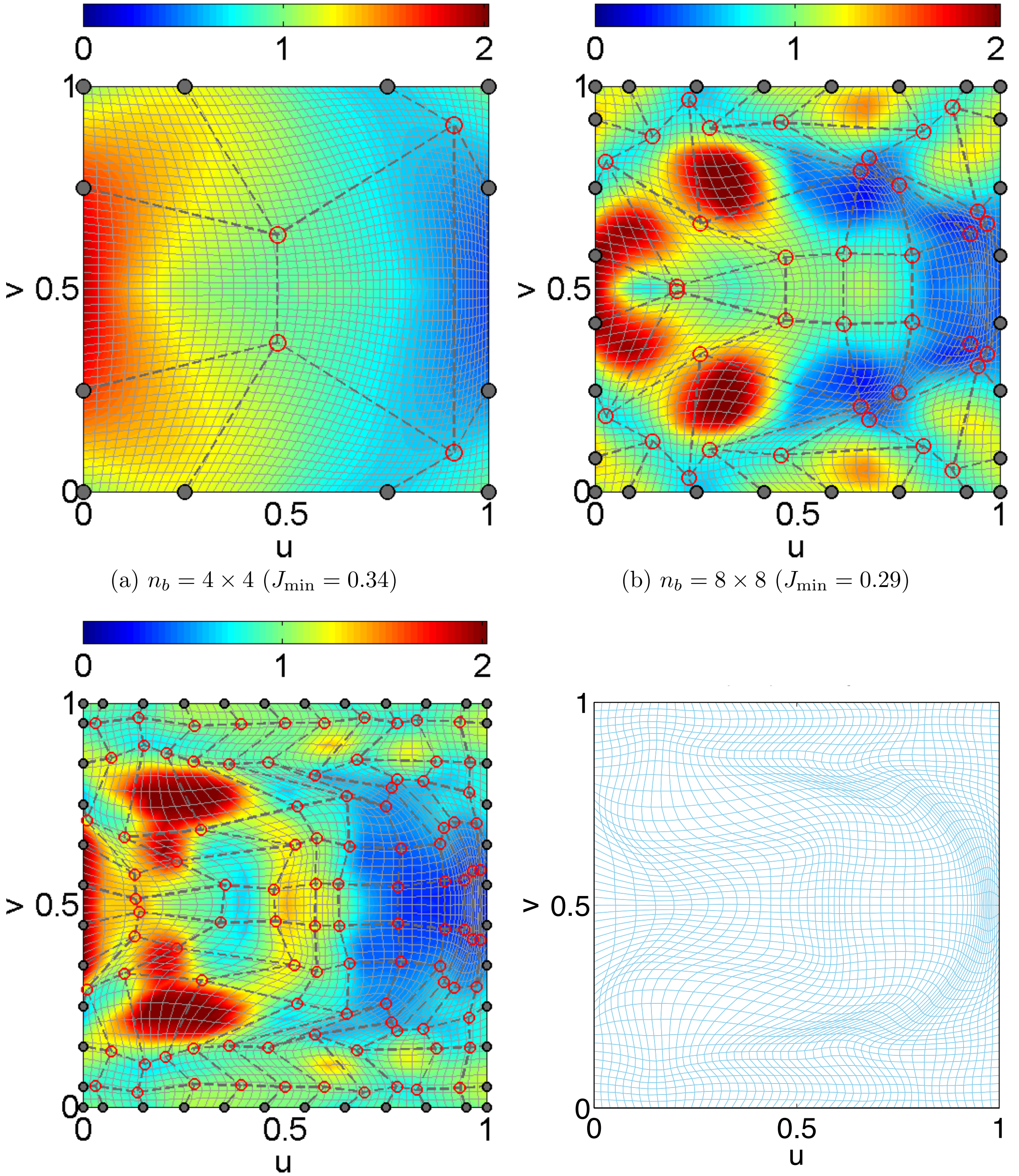


(a) Parametric grid under desired $\mathbf{D}_4^*(\mathbf{u})$ (b) Parametric grid from optimized $\mathbf{D}_4(\mathbf{u})$

Figure 2.13. Desired regular landmark grid deformation and parametric grid under optimized $\mathbf{D}_4(\mathbf{u})$ with concatenation of Clamped Plate Spline warps.

Optimized reparameterization function. The desired optimization obtained by the CPS concatenation reparameterization can also be achieved via the B-spline reparameterization approach. Figure 2.14(d) shows the deformed landmark grid (originally a regular 51×51 grid) with 12×12 B-spline coefficients, which demonstrates similar motion in Figure 2.13(b). The deformed landmark grid and control coefficients grid at different resolutions of 4×4, 8×8 and 12×12 overlapped by the Jacobian color field are plotted in Figure 2.14(a)(b)(c) respectively. The color plot indicates the Jacobian field on a fixed scale over $[0, 2]$; a color closer to red has a larger amount of stretch, whereas being closer to blue means a larger amount of compression.

Time cost comparison. The time cost of entire correspondence optimization at the three reparameterization B-spline resolutions, in comparison to that of the CPS concatenation reparameterization, is shown as the convergence curve of DL history w.r.t to iteration/optimization number in Figure 2.15(a) and w.r.t elapsed time in Figure 2.15(b). It can be seen that in order to reduce the DL objective



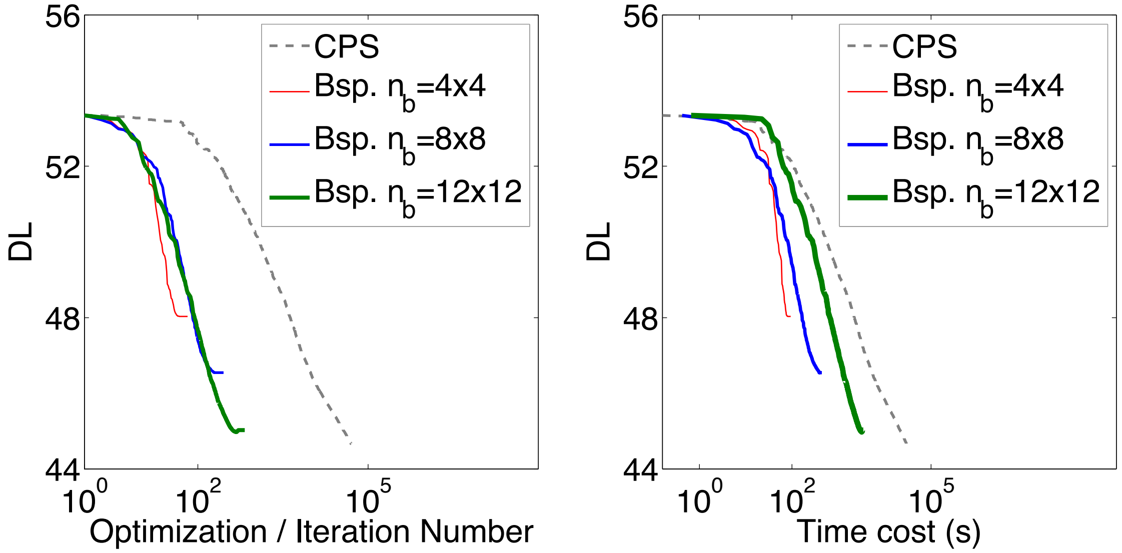
(a) $n_b = 4 \times 4$ ($J_{\min} = 0.34$)

(b) $n_b = 8 \times 8$ ($J_{\min} = 0.29$)

(c) $n_b = 12 \times 12$ ($J_{\min} = 0.25$)

(d) $n_b = 12 \times 12$, deformed grid

Figure 2.14. Optimized reparameterization $\mathbf{D}_4(\mathbf{u})$ for shape 4 obtained with different numbers of B-spline coefficients. The color field shows the Jacobian of $\mathbf{D}_4(\mathbf{u})$.



(a) DL w.r.t optimization (CPS) / iteration number (B-splines)

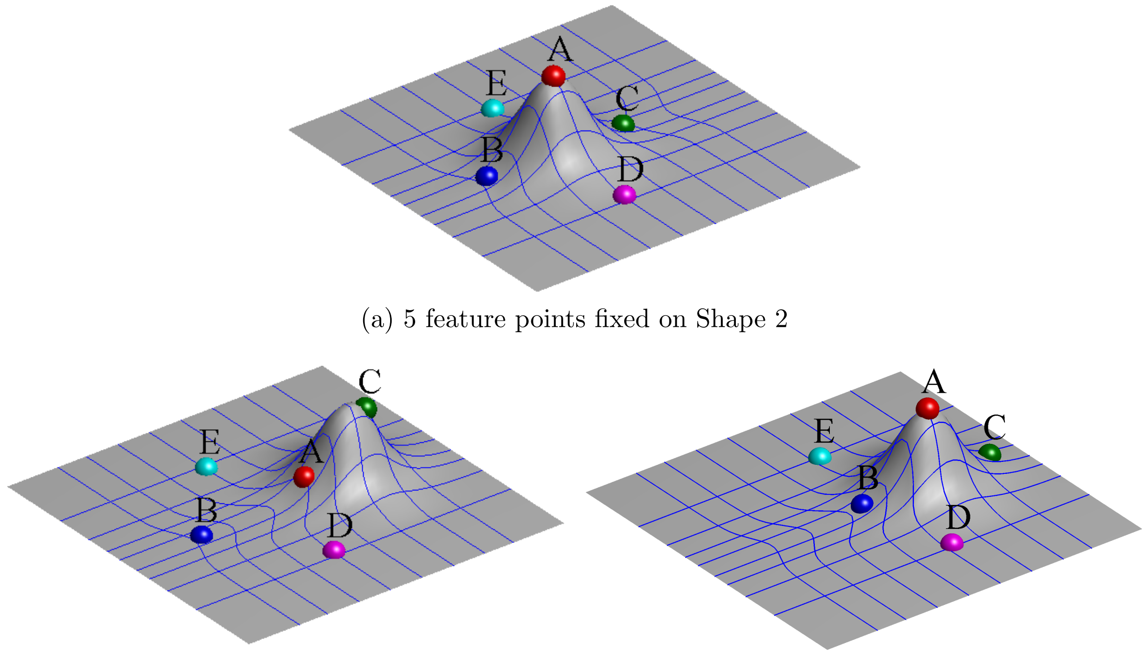
(b) DL w.r.t time cost

Figure 2.15. Time cost comparison: concatenation of CPS warps vs. direct reparameterization with B-splines.

function to the same level around 45, the direct reparameterization by B-spline with 12×12 coefficients takes 668 iterations and 3.4×10^3 seconds; it is about one order of magnitude faster than the concatenation of CPS approach which takes 5×10^4 iterations and 3×10^4 seconds.

Correspondence improvement after optimization. Figure 2.16 demonstrates the significant correspondence improvement with optimization. Before optimization, the five points on Shape 4 shown in Figure 2.16(b) corresponding to the five feature points fixed on the reference Shape 2 in Figure 2.16(a) indicate a poor initial correspondence (Point A signifies bump peak on Shape 2, but it lies almost at left bump bottom on Shape 4). After optimization with with $n_b = 12 \times 12$ reparameterization B-spline control coefficients, the five corresponding feature points on Shape 4 are brought very close to the expected feature locations as shown in Figure 2.16(c).

Constraint history and diffeomorphic conditions. Figure 2.17 shows the aggregated constraints of directly sampled Jacobians during the optimization



(a) 5 feature points fixed on Shape 2

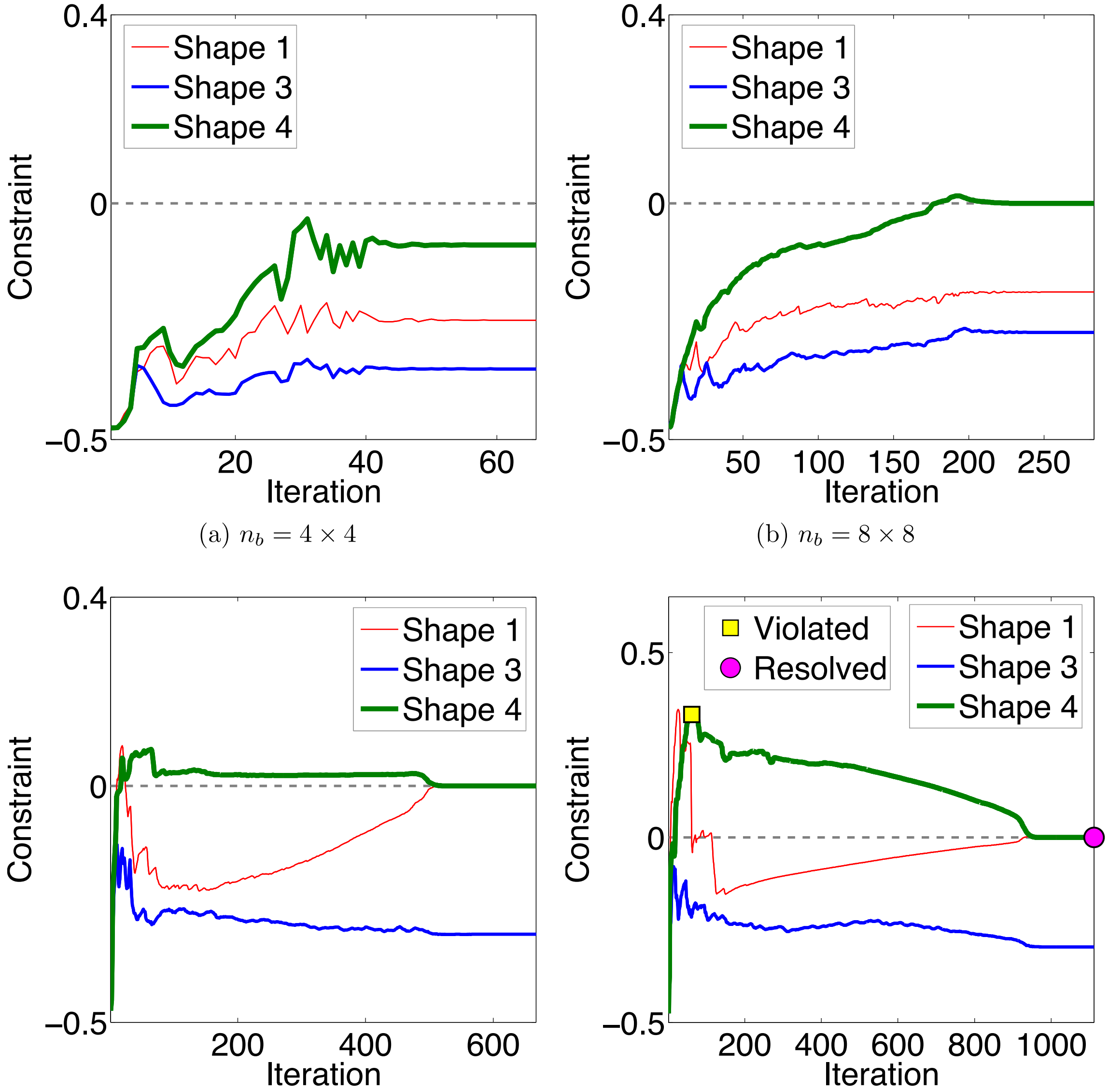
(b) Badly corresponded feature points on Shape 4 before optimization

(c) Well corresponded feature points on Shape 4 after optimization

Figure 2.16. Correspondence before and after optimization with $n_b = 12 \times 12$ control points in reparameterization B-spline.

iterations with different B-spline coefficients of 4×4, 8×8, 12×12 and 16×16. At the coarse representation ($n_b = 4 \times 4$), the 3 aggregated constraints for shapes 2, 3 and 4 are not active throughout the optimization process as shown in Figure 2.17(a) where all constraint values are negative. As the B-spline resolution for reparameterization increases to 8×8, 12×12 and 16×16, some of the aggregated Jacobian become acitive (i.e. positive) as shown in Figure 2.17(b)(c)(d). In the end, these violated constraints all converge to non-positive values, ensuring the diffeomorphism of the reparameterization.

In order to clearly demonstrate how the aggregated Jacobian constraints can guarantee a diffeomorphic reparameterization, we further increase the reparameterization B-spline resolution to 16×16. At resolution of 12×12 B-spline coefficients as shown in Figure 2.14(c), although the aggregated Jacobian violates the constraint but the original Jacobian is still positive and no self-intersection exists. As the reso-



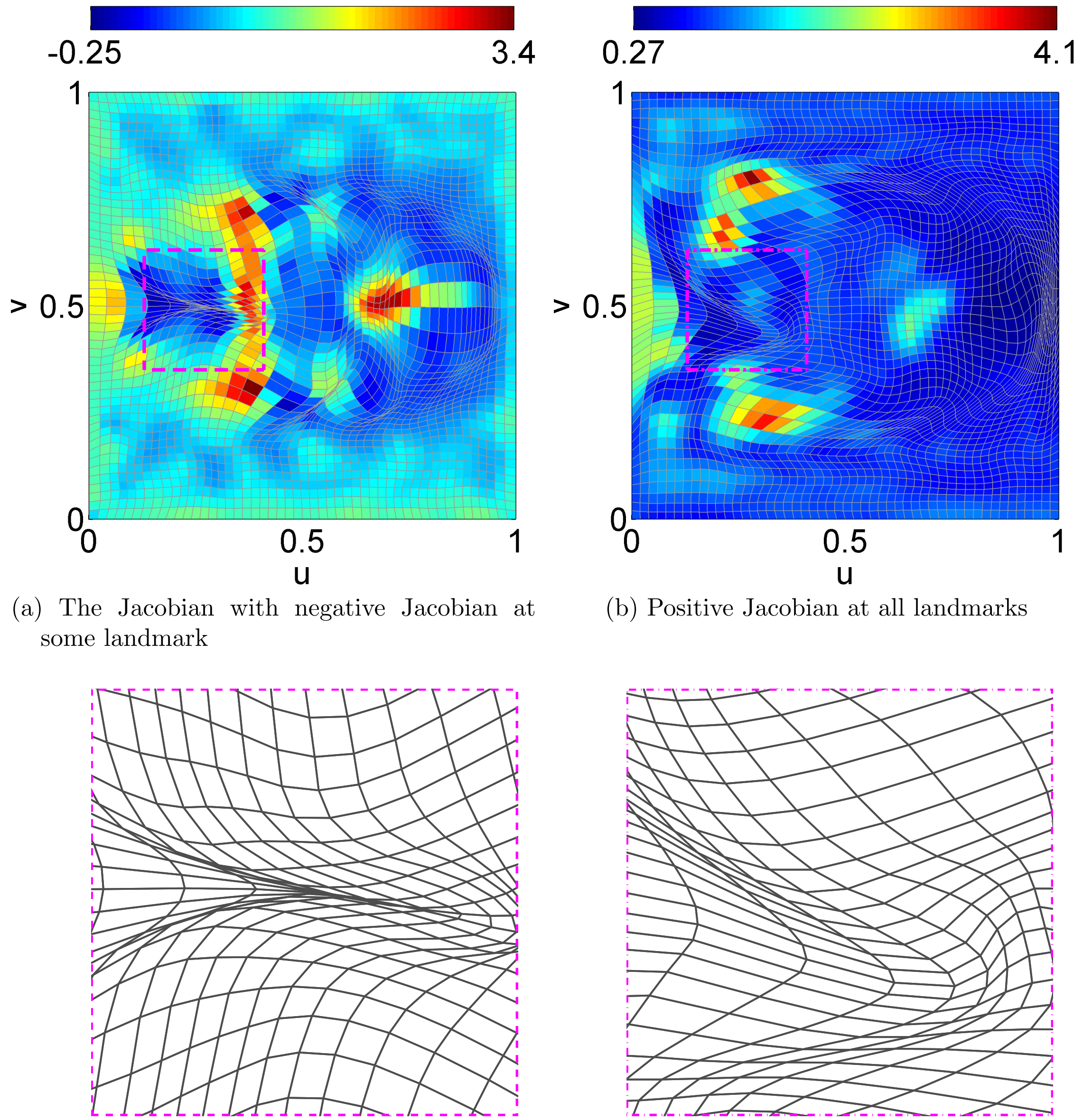
(a) $n_b = 4 \times 4$

(b) $n_b = 8 \times 8$

(c) $n_b = 12 \times 12$

(d) $n_b = 16 \times 16$: two iterations for Shape 4 picked with violated and resolved constraint in Figure 2.18

Figure 2.17. History of aggregated Jacobian constraint for the three non-reference shapes at four B-spline coefficient resolutions



(a) The Jacobian with negative Jacobian at some landmark

(b) Positive Jacobian at all landmarks

(c) Self-intersection in (a)

(d) No self-intersection in (b)

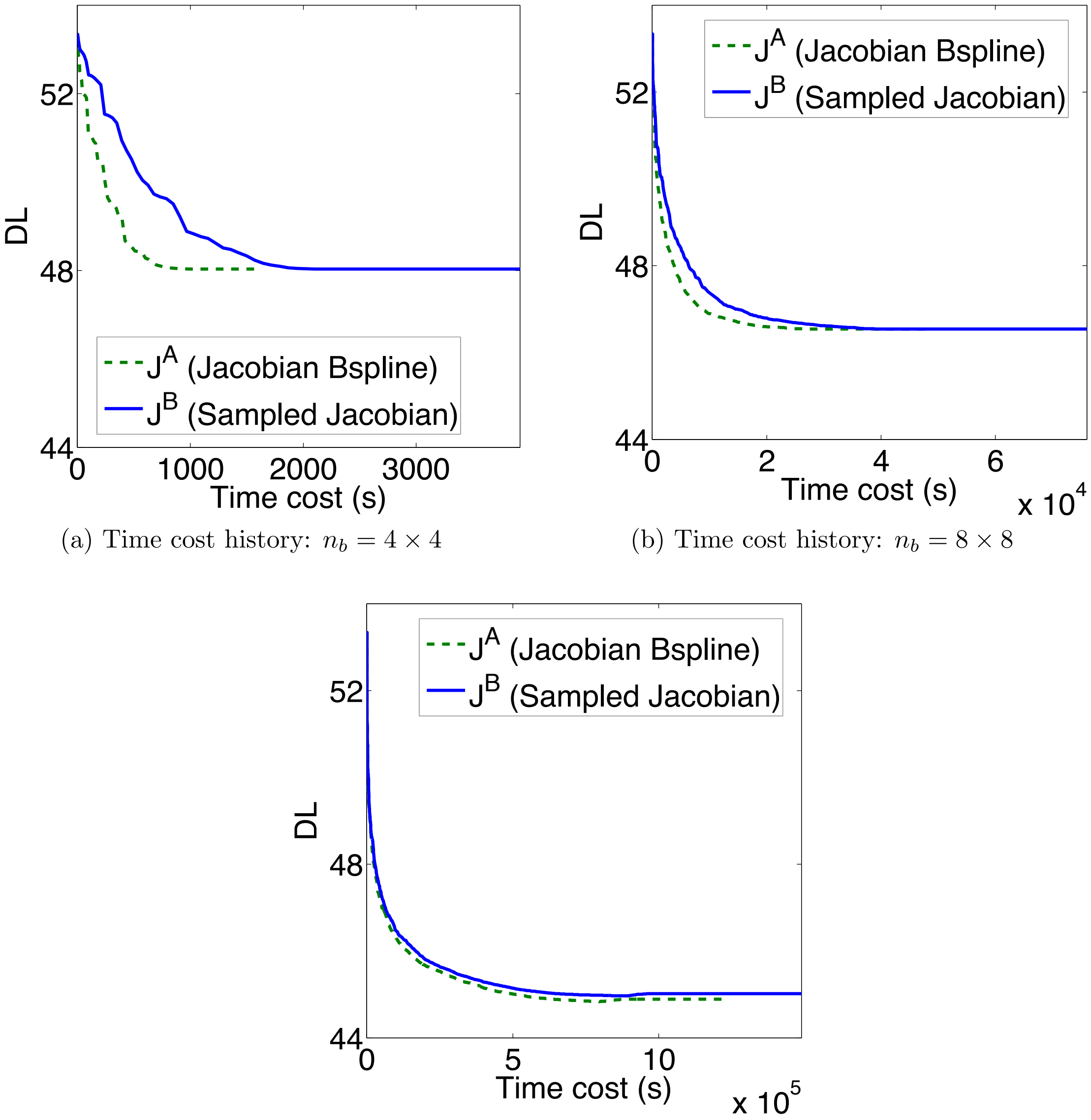
Figure 2.18. Diffeomorphism through Jacobian constraints in shape 4 with 16 × 16 B-spline coefficients

lution further increases to 16×16, the aggregated constraint is more severely violated and finally the original Jacobian drops below zero and self-intersection takes place. The aggregated constraints history for the three non-reference shapes are shown in Figure 2.17(d). We picked two particular iterations on the constraint history curve of Shape 4; one is where the aggregated constraint is most severely violated shown in square yellow marker called "Violated", and the other is the ending iteration with active constraint shown in red circle marker called "resolved".

The iteration with "violated" constraint is associated with the Jacobian field in Figure 2.18(a) with the aggregated constraint $J^{\text{aggrg}} = 0.33$ and the minimum Jacobian values at landmarks is $J_{\min}^{\text{raw}} = -0.25$ indicating a self-intersection at the vicinity of $\mathbf{u} = (0.2, 0.5)$ enclosed by the dotted square. After the optimization iterations, the Jacobian of the reparameterization from the final iteration is shown in Figure 2.18(b), and stops at an active aggregated constraint $J^{\text{aggrg}} = 4.019 \times 10^{-12}$, and the minimum Jacobian values for all 51×51 landmarks is $J_{\min}^{\text{raw}} = 0.27$. Figure 2.18(c) and (d) shows the zoom-in around $\mathbf{u} = (0.2, 0.5)$ of Figure 2.18(a) and (b) where there is initially self-intersection and it has since been untangled. This illustrates that when the aggregated Jacobian constraint becomes inactive, the resulting reparameterization is diffeomorphic.

Jacobian constraint comparison: Jacobian B-splines vs. sampled Jacobians. In addition to using the aggregated sampled Jacobians J^B in (2.11) as optimization constraint $\{g_l(\mathbf{b})\}$ via (2.22), the aggregated Jacobian B-spline option with J^A as defined in (2.10) is also implemented for comparison. The time cost history between the two constraint options are shown in Figure 2.19, where slightly better computational efficiency with J^A is observed.

In lightly of the ease of implementation using J^B and the small difference in the optimization results, the sampled Jacobian positivity constraint option is adopted



(a) Time cost history: $n_b = 4 \times 4$

(b) Time cost history: $n_b = 8 \times 8$

(c) Time cost history: $n_b = 12 \times 12$

Figure 2.19. Comparison of optimization objective and constraint between constraint option A and B

for all the 3D experiments in this thesis.

2.3.3 Computational analysis of analytical gradient.

Section 2.2.2 introduces the analytical gradient formula in the form of direct and adjoint sensitivities. Now we use the concrete examples of 2D box-bump and 3D plane-bump presented above to demonstrate the accuracy, correctness and efficiency of analytical gradient by comparing it with numerical approximation. We also show adjoint sensitivity formula outperforms direct sensitivity in terms of computational cost w.r.t resolution parameters and shape population scale, making adjoint the preferred formula for implementing analytical gradient.

2.3.3.1 Analytical gradient vs. numerical approximation.

Within the context of the 3D plane-bump example in Section 2.3.2, the objective function gradients $\frac{\mathrm{d}\, f(\mathbf{b})}{\mathrm{d}\, b_r}$ using analytical formula can be compared with numerical approximation in terms of results accuracy and time cost. The analytical equation refers to the direct sensitivity in (2.15) (2.16) and the adjoint sensitivity in (2.20) (2.21). The numerical approximation uses finite difference method

$$\frac{\mathrm{d}\, f(\mathbf{b})}{\mathrm{d}\, b_r} \approx \frac{f(\mathbf{b}+h\mathbf{e}_r)-f(\mathbf{b})}{h}$$

where the $n_b \times 1$ unit vector $\mathbf{e}_r = [0, ..., 0, 1, 0, ..., 0]^T$ has “1” is in the r-th row, and the small step length is $h = 10^{-6}$.

Table 2.1 compares the gradients results between direct and adjoint sensitivity along with numerical aproximation. The landmark resolution is $n_P = 51 \times 51$, and reparametrization B-spline resolution is $n_b = 4 \times 4$ as seen in Figure 2.20(a), resulting in $n_D = 24$ optimization variables and objective function gradients. The table shows 8 gradient values out of 24 for all three evaluation options, when the optimization variables $\{b_r\}(r = 1, ..., 24)$ are randomly selected. It’s shown that the two analytical sensitivities are very close to each other (consistent through the 9th digit), and numerical approximation gives slightly off accuracy (consistent only till the 3rd digit) due to the difficulty of determining a good step length.

Table 2.1. Gradient results comparison between analytical (direct and adjoint sensitivity) and numerical (finite difference) approach. Landmark resolution $n_P = 51 \times 51$, reparametrization B-spline coefficients resolution $n_b = 4 \times 4$ as shown in Figure 2.20(a).

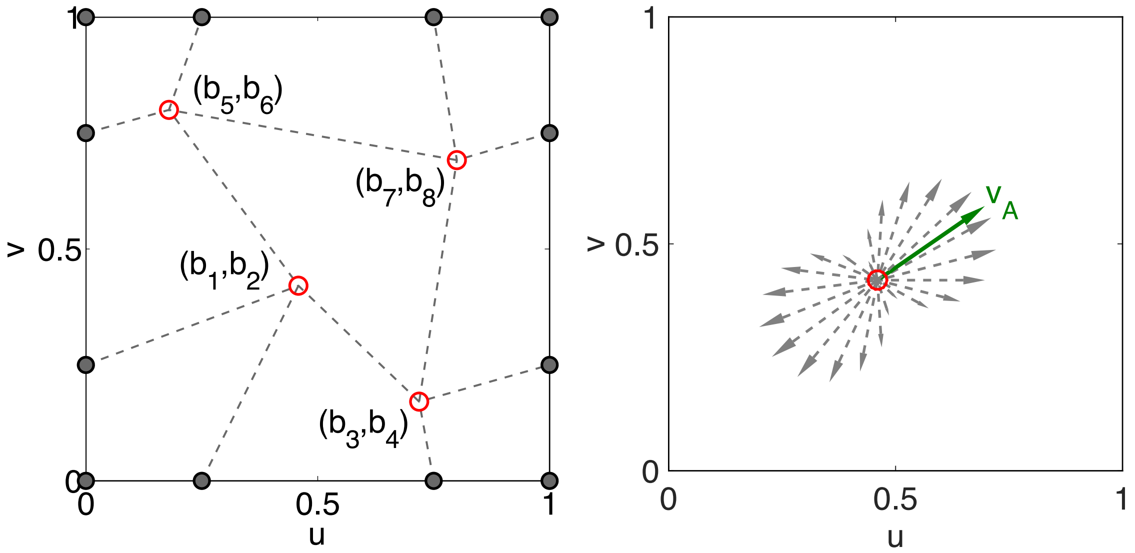
$\frac{\mathrm{d}f}{\mathrm{d}b_r}$	Direct	Adjoint	Numerical
$r = 1$	3.9459884974	3.9459884974	3.9459915797
$r = 2$	2.7229409584	2.7229409585	2.7229473574
$r = 5$	1.3655139494	1.3655139494	1.3655253781
$r = 9$	2.5544258045	2.5544258045	2.5545317968
$r = 13$	0.9374377692	0.9374377692	0.9375618646
$r = 17$	-2.6814822821	-2.6814822821	-2.6814423464
$r = 20$	0.2758634880	0.2758634880	0.2759022806
$r = 24$	-0.2397113282	-0.2397113282	-0.2396724241

Table 2.2 shows the time cost comparison between analytical and numerical gradient computations. The reparametrization B-spline resolutions are 4 × 4, 8 × 8, 12 × 12 and 16 × 16. It's seen that adjoint is the fastest, direct is slightly slower and numerical approximation is the slowest; the computational efficiency advantage is particularly noticeable as reparametrization B-spline is refined.

2.3.3.2 Validation by directional derivative. Previously it was shown that analytical and numerical gradients are consistent. In order to further test the correctness of the analytical gradients, we compare the analytical gradient couplet with the numerical approximated steepest ascending direction. Taking again the current example for instance, the 24 optimization variables determines the (u, v) positions of the 12 reparamerization B-spline control points in the parameter domain, where each of Shape 1, 3 and 4 has 4 control coefficients points. Figure 2.20(a) shows the

Table 2.2. Gradient evaluation time cost comparison between analytical (direct and adjoint sensitivity) and numerical (finite difference) approach. Landmark resolution $n_P = 51 \times 51$, reparametrization B-spline coefficients resolution are $n_b = 4 \times 4$ in as Figure 2.18(a), $n_b = 8 \times 8$ as in Figure 2.18(b), $n_b = 12 \times 12$ as in Figure 2.18(c) and $n_b = 16 \times 16$.

Time cost (s)	Direct	Adjoint	Numerical
$n_b = 4 \times 4$	0.1170	0.1098	1.3848
$n_b = 8 \times 8$	0.1284	0.1225	7.6627
$n_b = 12 \times 12$	0.2566	0.2329	24.7603
$n_b = 16 \times 16$	0.4062	0.3719	56.8001



(a) The 8 design variables $b_1, b_2, ...b_8$ control the reparametrization B-spline on Shape 1

(b) Analytical gradient $\mathbf{v}_A$ (green) and numerical gradient (closest dotted to green) identified as the maximum directional derivative out of all directions (dotted arrows), where $n_V = 25$ directions.

Figure 2.20. The 4 coefficient points determined by the 8 design variables on Shape 1 and the analytical with numerical gradient at the 1st coefficient point

4 coefficient points $(b_1, b_2), (b_3, b_4), (b_5, b_6), (b_7, b_8)$ associated with Shape 1. These 4 points along with the remaining 8 are randomly placed in the parameter domain for testing purpose.

We use the 1st coefficient point (b_1, b_2) of Shape 1 for illustration. The analytical gradient at reparametrization B-spline's coefficient point $\mathbf{p}$ is obtained by adjoint formula as $\mathbf{v}_A = \left[\frac{\mathrm{d}\,f(\mathbf{p})}{\mathrm{d}\,b_1}, \frac{\mathrm{d}\,f(\mathbf{p})}{\mathrm{d}\,b_2}\right]^T = \left[\frac{\mathrm{d}\,f(\mathbf{b})}{\mathrm{d}\,b_1}, \frac{\mathrm{d}\,f(\mathbf{b})}{\mathrm{d}\,b_2}\right]^T = [3.9460, 2.7229]^T$.

Then we subdivide the full 360° range into n_V angle intervals, yielding the same number candidate directions $\mathbf{v}_i = \left[\cos\frac{2\pi(i-1)}{n_V}, \sin\frac{2\pi(i-1)}{n_V}\right]^T$. The directional derivative at $\mathbf{p}$ along direction $\mathbf{v}_i$ is approximated by finite difference

$$\nabla_{\mathbf{v}_i} f(\mathbf{p}) \approx \frac{f(\mathbf{p} + h\mathbf{v}_i) - f(\mathbf{p})}{h}$$

. The direction $\mathbf{v}^*$ with the maximum directional derivative is just the steepest ascending direction and its gradient magnitude is $\nabla_{\mathbf{v}^*} f(\mathbf{p})$, making the numerical gradient $\mathbf{v}_N = \mathbf{v}^* \nabla_{\mathbf{v}^*} f(\mathbf{p})$. As an example, when the full circle is equally subdivided into $n_V = 25$ directions as shown in Figure 2.20(b), the directional derivatives are plotted in arrows pointing in their directions and with positive magnitude in solid and negative magnitude in dotted line. The direction with the largest positive directional derivative is identified as the numerical gradient shown in blue, which is fairly close to the analytical gradient computed by adjoint formula shown in green.

Table 2.3 shows the analytical gradient quantitatively validated by the numerical gradient as the numerical resolution n_V increases. The row of $n_V = 25$ corresponds to the scenario in Figure 2.20(b). It is observed that the numerical results tend toward the analytical direction as number of angle intervals increases. Due to random nature of the position of selected coefficient point, the correctness of the analytical gradient obtained via proposed adjoint method has been tested successfully. Similar validation can be carried out for the remaining 11 coefficient points $[b_3, b_4]^T, [b_5, b_6]^T, ..., [b_{23}, b_{23}]^T$.

2.3.3.3 Direct vs. adjoint sensitivity. The analytical gradients obtained by direct and adjoint formula have been tested after comparison with finite difference based numerical approximation, with regard to both results and correctness. Both

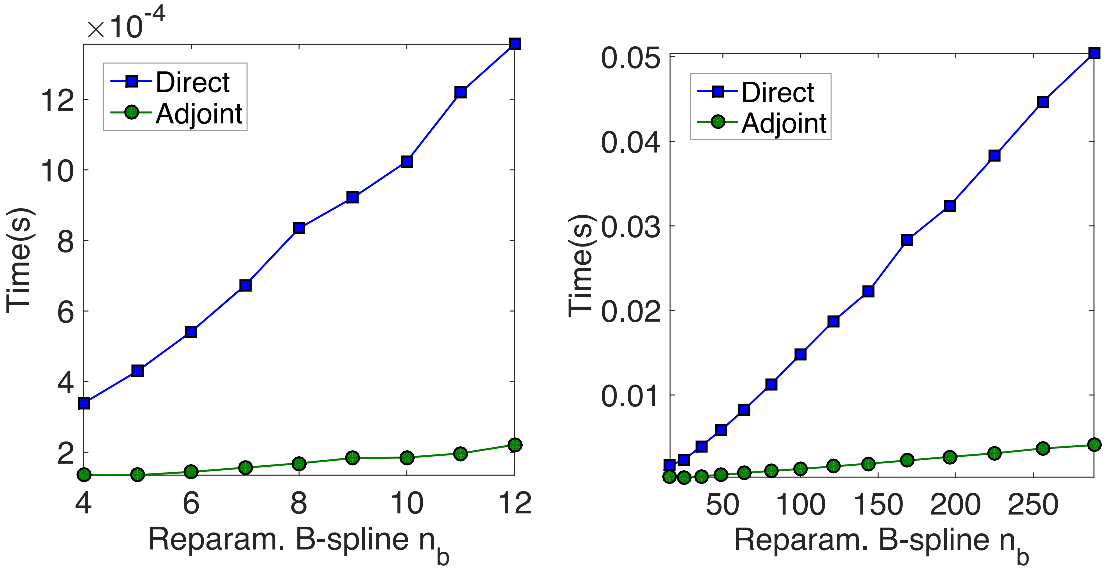
Table 2.3. Analytical gradient $\mathbf{v}_A = \left[\frac{\mathrm{d}\,f(\mathbf{b})}{\mathrm{d}\,b_1}, \frac{\mathrm{d}\,f(\mathbf{b})}{\mathrm{d}\,b_2}\right]^T$ compared with numerical approximation by directional derivatives $\mathbf{v}_N = \mathbf{v}^* \nabla_{\mathbf{v}^*} f(\mathbf{p})$.

Option		Gradient	
Numerical (Finite Difference)	$n_V = 25$	4.1819	2.2927
	$n_V = 100$	3.8808	2.8129
	$n_V = 250$	3.9513	2.7152
	$n_V = 500$	3.9513	2.7152
	$n_V = 1000$	3.9513	2.7152
	$n_V = 2500$	3.9445	2.7251
	$n_V = 5000$	3.9445	2.7251
Analytical (Adjoint formula)		3.9460	2.7229

the direct and adjoint formulas produce the same gradient results, however, their efficiency performance differ in general situations. Now the 2D box-bump example and 3D plane-bump example are chosen to demonstrate their difference in computational efficiency, and we compare the time cost, for each iteration, of computing the gradient of the description length with respect to B-spline coefficients $\mathbf{b}$ by the direct and adjoint method. The situations are discussed where each one varies only one parameter with all the other parameters held fixed.

Figure 2.23(a) shows the computing time for 2D training set called box-bump in Figure 2.5 and Figure 2.23(b) for 3D data plane-bump in Figure 2.12. It can be seen that the adjoint method is more efficient in computing the gradient when the number of shape instances become large for both 2D and 3D. In light of this significant computational speed gain, the adjoint sensitivity is the method used in the subsequent numerical implementation.

Figure 2.21 shows the influence of reparametrizatoin B-spline coefficients resolution n_b on computation time when using direct and adjoint formulas. It's observed that the adjoint sensitivity enjoys a very small time increase as reparametrization B-spline is refined. Here the landmark resolution is fixed at $n_P = 51$ for 2D box-bump and $n_P = 51 \times 51 = 2601$ for 3D plane-bump, and shape number is fixed at $n_S = 4$.



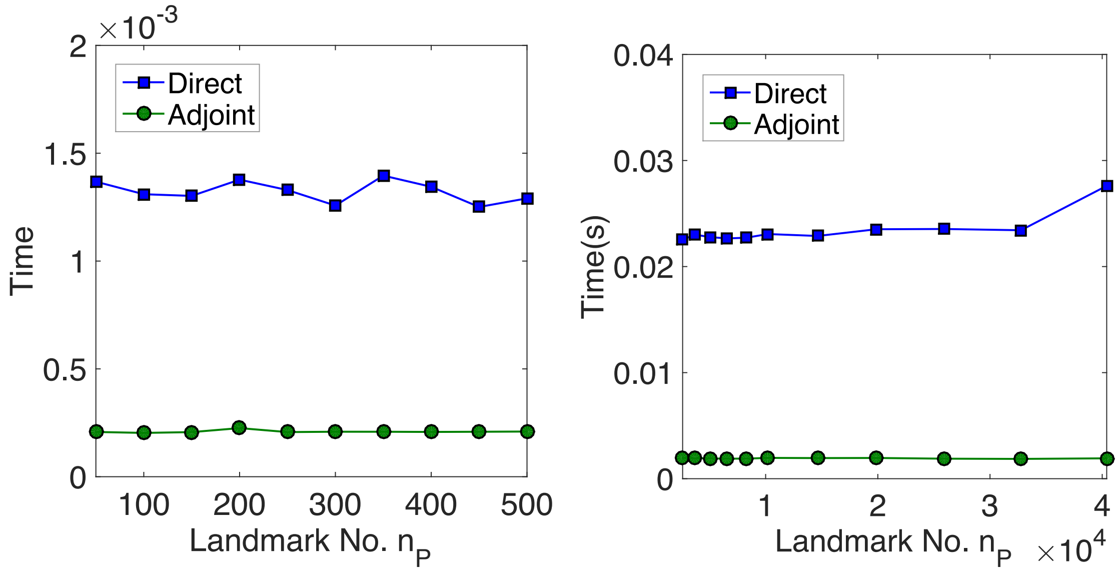
(a) 2D box-bump: vary n_b, with $n_P = 51, n_s = 4$ fixed

(b) 3D plane-bump: vary $n_b = n_{b1}n_{b2}$, with $n_P = 2601(n_{P1} = n_{P2} = 51), n_S = 4$ fixed

Figure 2.21. Computing time for direct sensitivity and adjoint sensitivity w.r.t reparametrization B-spline coefficients resolution n_b at each iteration for both curve and surface training sets.

Figure 2.22 shows the influence of landmark resolution n_P on computation time when using direct and adjoint formulas. It's observed that the adjoint formula takes much shorter time as landmark grid is refined. Here the reparametrization B-spline resolution is fixed at $n_b = 12$ for 2D box-bump and $n_b = 12 \times 12 = 144$ for 3D plane-bump, and shape number is fixed at $n_S = 4$.

Figure 2.23 shows the influence of shape population size (shape number) n_S on computation time when using direct and adjoint formulas. It's observed that the adjoint formula is very insensitive to shape number and makes it possible for the algorithm to scale to large population. Here the reparametrization B-spline resolution



(a) 2D box-bump: vary n_P, with $n_S = 4, n_b = 12$ fixed

(b) 3D plane-bump: vary $n_P = n_{P1}n_{P2}$, with $n_S = 4, n_b = 144(n_{b1} = n_{b2} = 12)$ fixed

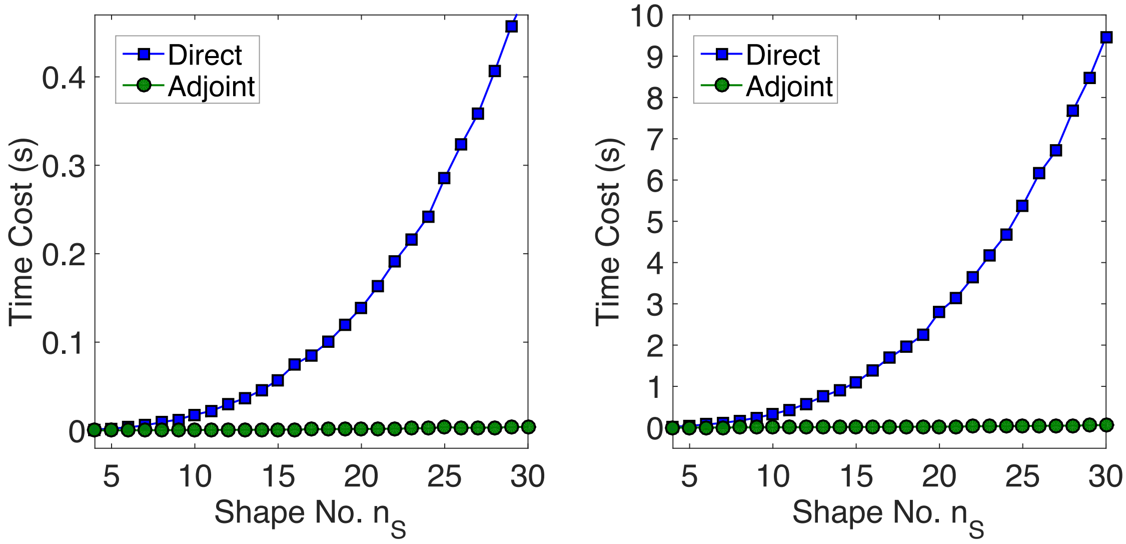
Figure 2.22. Computing time for direct sensitivity and adjoint sensivity w.r.t landmark number n_P at each iteration for both curve and surface training sets.

is fixed at $n_b = 12$ for 2D box-bump and $n_b = 12 \times 12 = 144$ for 3D plane-bump, and the landmark resolution is fixed at $n_P = 51$ for 2D box-bump and $n_P = 51 \times 51 = 2601$ for 3D plane-bump.

From these three circumstances, it's clear that although direct and adjoint sensitivity yield same gradient results, the adjoint method outperforms direct formula due to its superior computational efficiency advantage.

2.3.4 2D real data: Hand.

2.3.4.1 Training set B-spline pre-processing. We test our correspondence optimization on a real data set from 40 raw images [56]. One example image is shown in Figure 2.24(a). A quadratic B-spline is fit to each raw image of the hand contours and the control points and knots of a B-spline curve are shown in Figure 2.24(b). The number of control points vary from 50 to 63 and generally different from one to another due to the slightly different shapes in different images. All the 40 B-spline



(a) 2D box-bump: vary n_S, with $n_P = 51, n_b = 12$ fixed

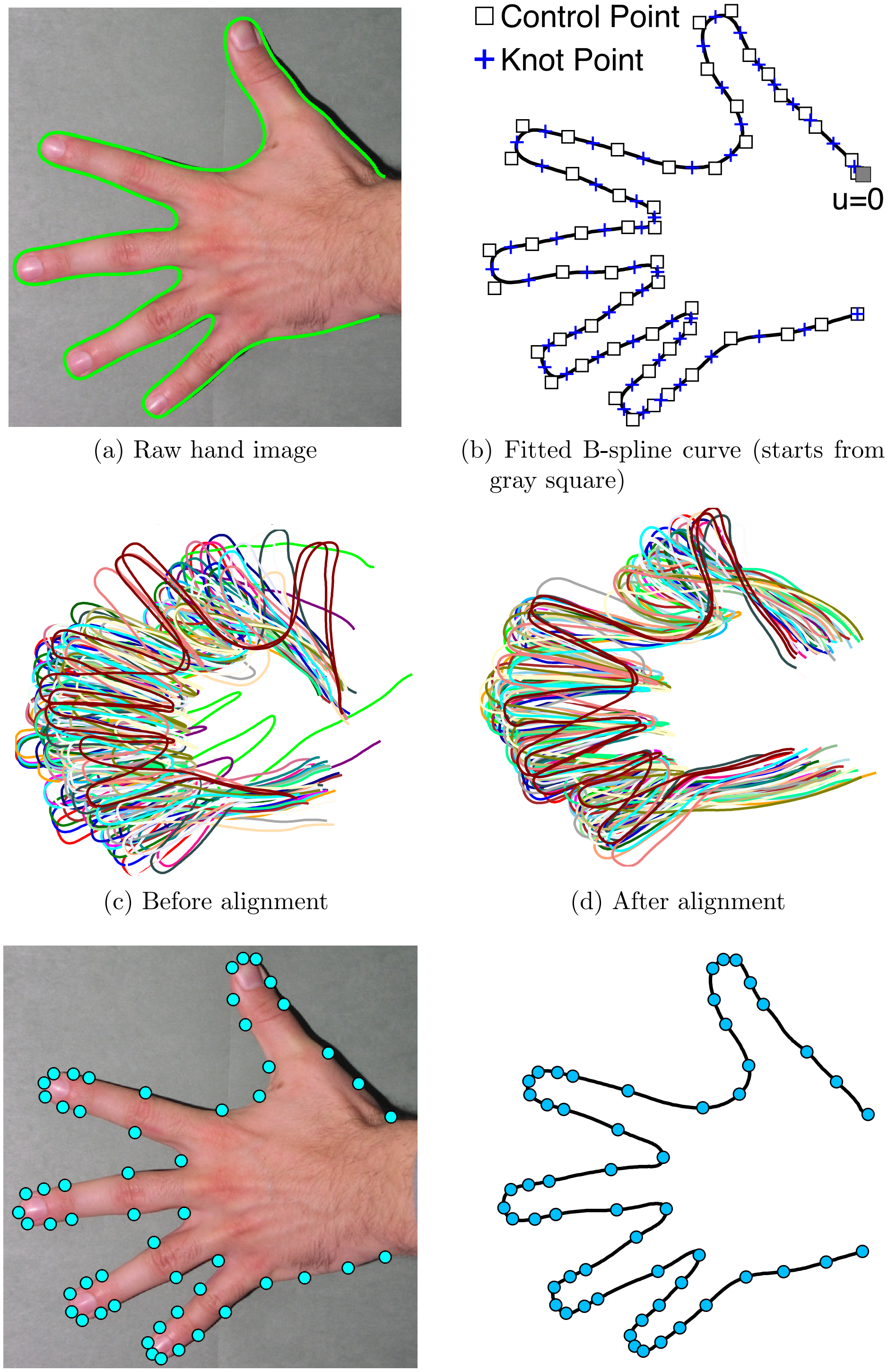
(b) 3D plane-bump: vary n_S, with $n_P = 2601 (n_{P1} = n_{P2} = 51), n_b = 144 (n_{b1} = n_{b2} = 12)$ fixed

Figure 2.23. Computing time for direct sensitivity and adjoint sensivity w.r.t shape number n_S at each iteration for both curve and surface training sets.

represented hands are superimposed in Figure 2.24(c). Since the generalized Procrustes Analysis involves iterative procedure and drastically slows down the entire correspondence optimization if it is included, the shape alignment is performed only once before the actual optimization. This practice further improves the computation efficiency and is observed to have little influence on the result compared to including alignment during every optimization iteration. The one-time alignment result is shown in Figure 2.24(d).

2.3.4.2 Optimization via reparameterization B-spline.

Optimized reparameterization function and correspondence improvement. The reparameterization function is parameterized with 12 B-spline coefficients. The landmark resolution is $n_P = 151$ and landmarks are sampled uniformly in the parameter domain before optimization. Shape 1 is selected as the reference shape. For brevity, only Shape 21 is picked out of the remaining 39 shapes to show the correspondence change before and after optimization.



(a) Raw hand image

(b) Fitted B-spline curve (starts from gray square)

(c) Before alignment

(d) After alignment

(e) Feature points on image

(f) Feature points on B-spline

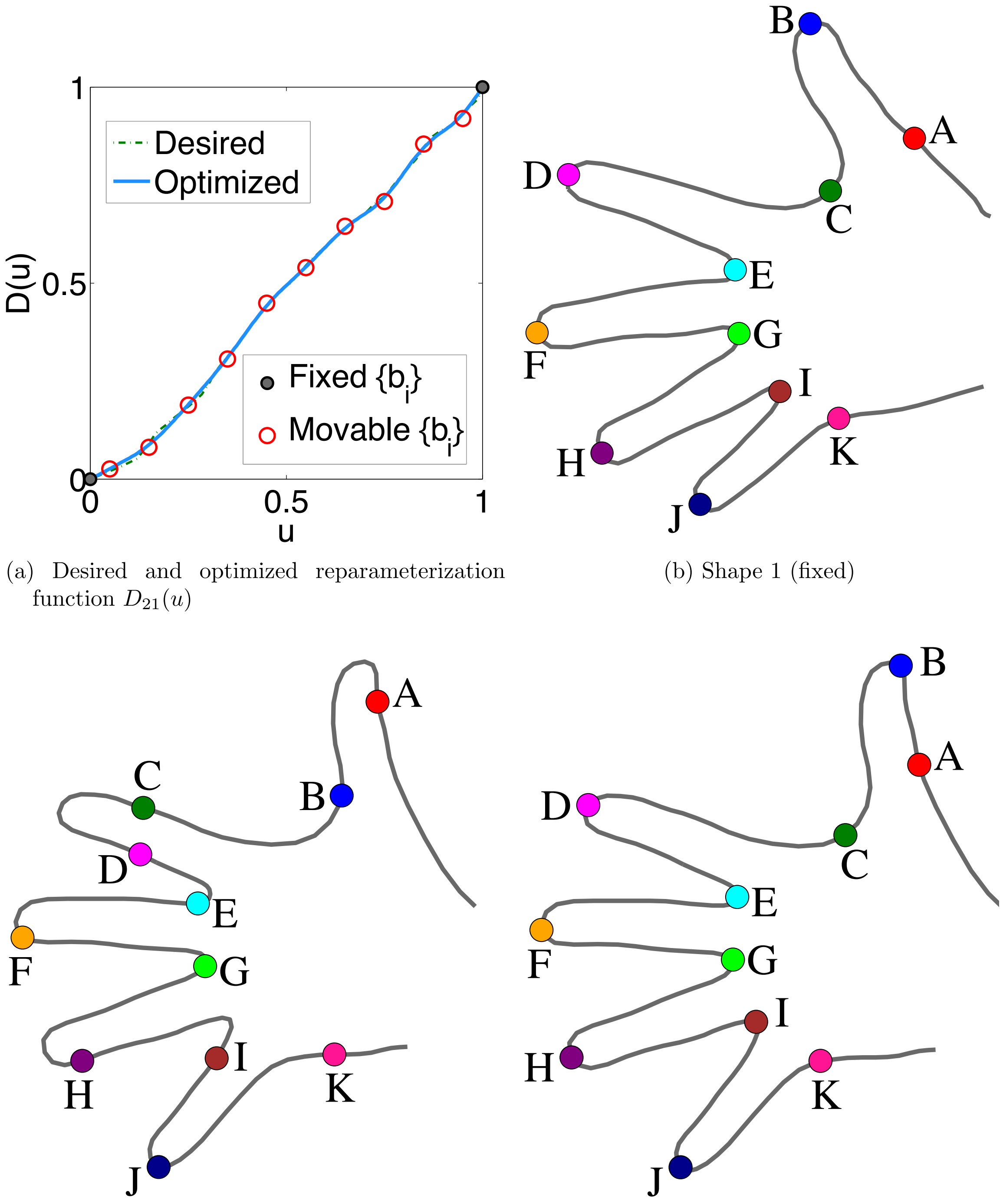
Figure 2.24. Pre-processing the training set of hand data.

The initial reparameterization function for optimization is the identity function. To obtain the desired reparameterization function for comparison, the original, manually marked 56 raw feature points accompanying the data set in [56] are used to determine those parameters of the detailed feature points of all instances. The 56 features points are plotted on one raw image and the corresponding B-spline curve in Figure 2.24(e) and (f), and they include the finger tips, gaps and knuckle points. Comparing the 56 corresponding parameters of all the 40 instances lead to the 39 desired reparameterization functions.

The initial, desired , and optimized reparameterization functions of Shape 21 are then plotted in Figure 2.25(a). It can be seen that although the desired function is only a manual approximation under the guide of basic hand anatomy knowledge, the optimized correspondence matches with it very well. All the other 38 shapes exhibit this similar results although not shown here.

Out of the 56 original feature points from the images, we plot 11 key feature points for a more clear demonstration of correspondence improvement due to the optimization. The 11 feature points located at finger tips and valleys on the reference Shape 1 is shown in Figure 2.25(b). Before optimization the corresponding points on Shape 21 that have the same parameters as those on Shape 1 are situated at the positions displayed in Figure 2.25(c), where most feature points are far away from the finger tips/valleys where they are expected to be. After optimization, they are brought to the locations in good correspondence to the feature points on Shape 1 as shown in Figure 2.25(d).

Statistical modes improvement. Figure 2.26 and Figure 2.27 respectively show the first two modes of variations from the statistical models with and without correspondence optimization. It can be seen that the shape variation without the optimization is highly improbable: the middle and index fingers are all out of proportion



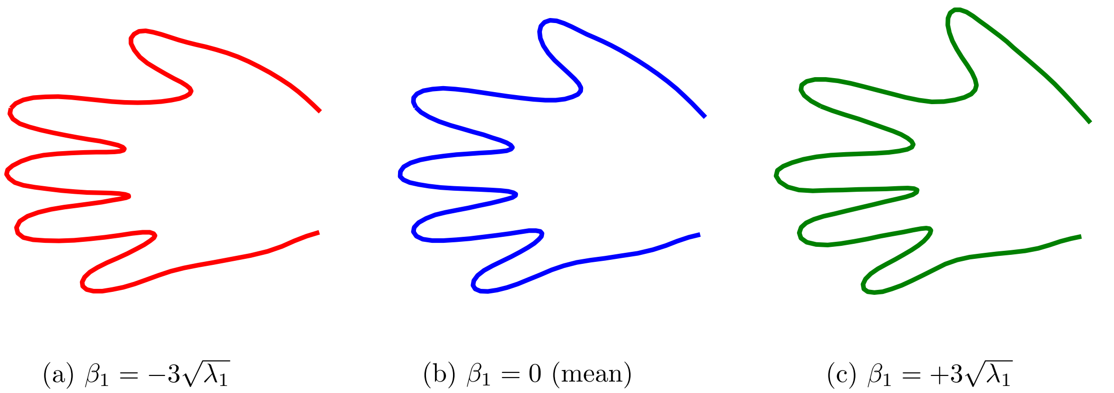
(a) Desired and optimized reparameterization function $D_{21}(u)$

(b) Shape 1 (fixed)

(c) $\mathbf{S}_{21}(u)$

(d) Optimized: $\mathbf{S}_{21}(D(u))$

Figure 2.25. Feature points on Shape 21 before and after optimizing correspondence with respect to shape 1.


(a) $\beta_1 = -3\sqrt{\lambda_1}$ (b) $\beta_1 = 0$ (mean) (c) $\beta_1 = +3\sqrt{\lambda_1}$

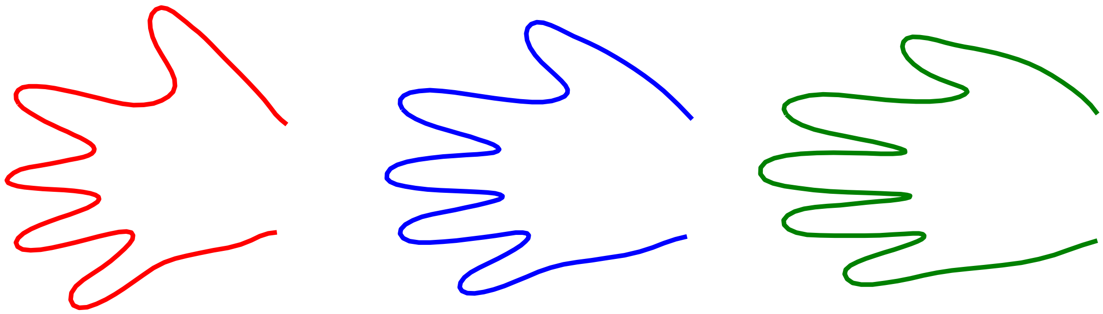

(d) $\beta_2 = -3\sqrt{\lambda_2}$ (e) $\beta_2 = 0$ (mean) (f) $\beta_2 = +3\sqrt{\lambda_2}$

Figure 2.26. The first 2 modes of statistical model for hand data before optimization. $\lambda_1 = 723.2(31.48\%), \lambda_2 = 467.7(20.36\%)$

in length and width. The first mode of the statistical model before correspondence optimization is only 31.48% of the total variation. The shape variation from the optimized correspondence looks realistic. The first mode of the statistical model with correspondence optimization is 50.24% of the total variation.

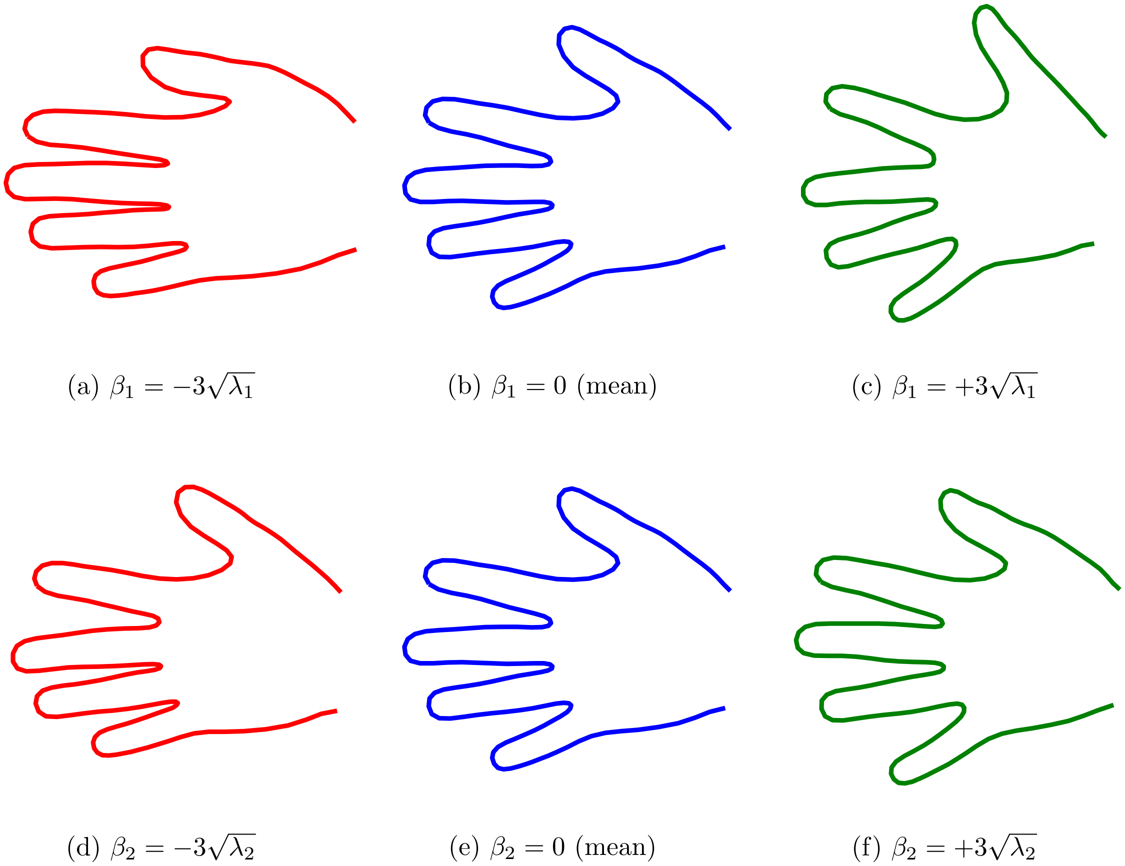
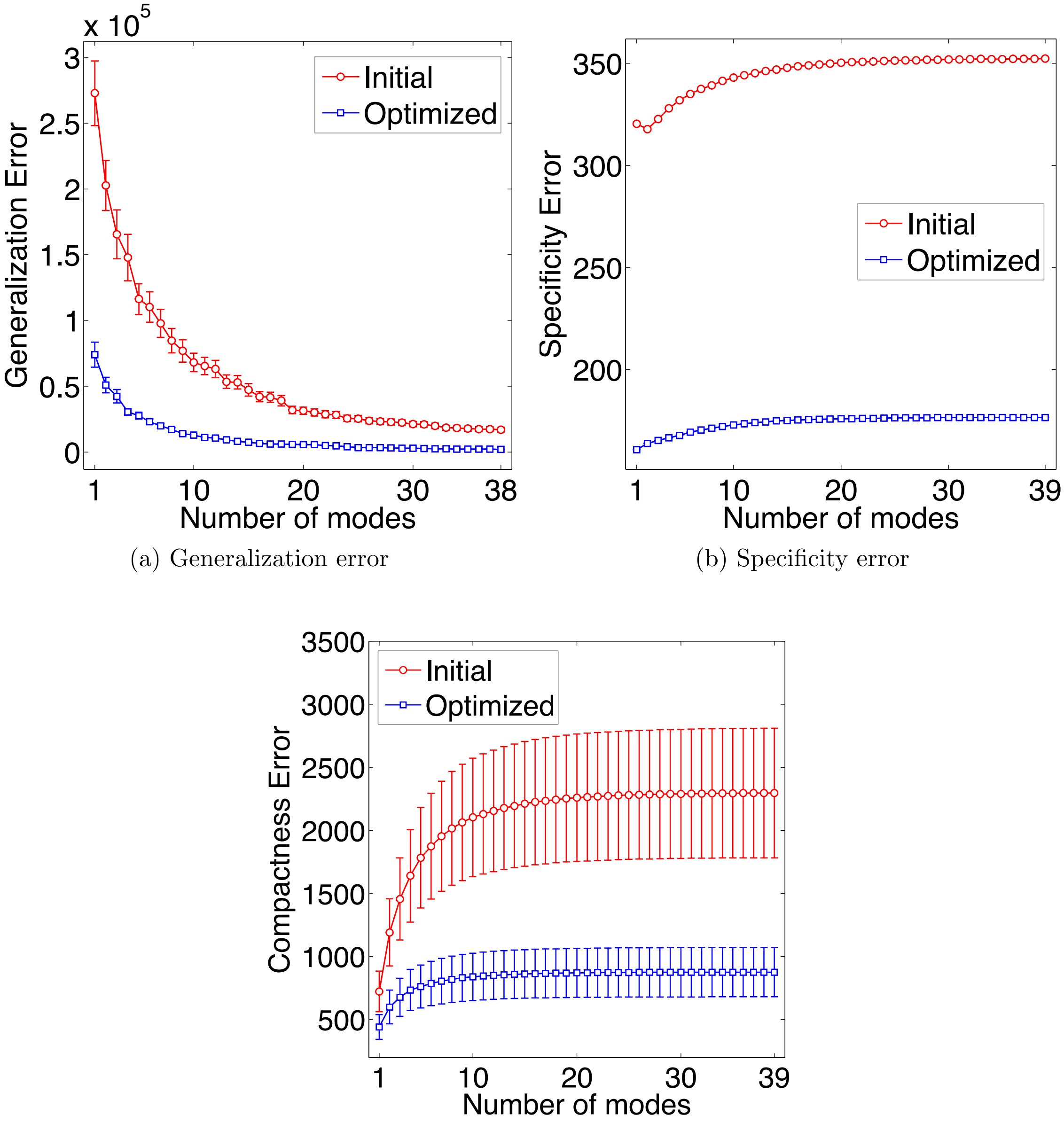

(a) $\beta_1 = -3\sqrt{\lambda_1}$
(b) $\beta_1 = 0$ (mean)
(c) $\beta_1 = +3\sqrt{\lambda_1}$
(d) $\beta_2 = -3\sqrt{\lambda_2}$
(e) $\beta_2 = 0$ (mean)
(f) $\beta_2 = +3\sqrt{\lambda_2}$

Figure 2.27. The first 2 modes of statistical model for hand data after optimization. $\lambda_1 = 438.4(50.24\%), \lambda_2 = 153.44(18.04\%)$

Such differences in shape variations from the models are a direct consequence of the optimized correspondence from Figure 2.25(c) to Figure 2.25(d) as compared to Figure 2.25(b).

SSM evaluation measure improvement. To further compare the quality improvement of the statistical model before and after optimization, the three quantitative measures of SSM proposed in [28] are computed. Figure 2.28 compares the



(a) Generalization error

(b) Specificity error

(c) Compactness error

Figure 2.28. SSM quality before and after optimization

generalization ability, specificity and compactness error [28] before and after correspondence optimization.

The quantitative measure of the generalization ability is achieved by the leave-one-out procedure, where one shape is chosen to compare with its reconstructed shape by the statistical model built from the remaining $n_S - 1$ shapes. More specifically, the reconstructed shape $\tilde{\mathbf{x}}_i$ for the i-th shape $\mathbf{x}_i$ can be obtained using (1.9). Varying the number of modes n_m for the linear approximation, the leave-one-out generalization measure of SSM is given by

$$E_G(n_m) = \frac{1}{n_S} \sum_{i=1}^{n_S} ||\mathbf{x}_i - \tilde{\mathbf{x}}_i||. \tag{2.25}$$

The standard error of generalization measure is defined by

$$\sigma_G(n_m) = \frac{\sigma}{\sqrt{n_S - 1}}. \tag{2.26}$$

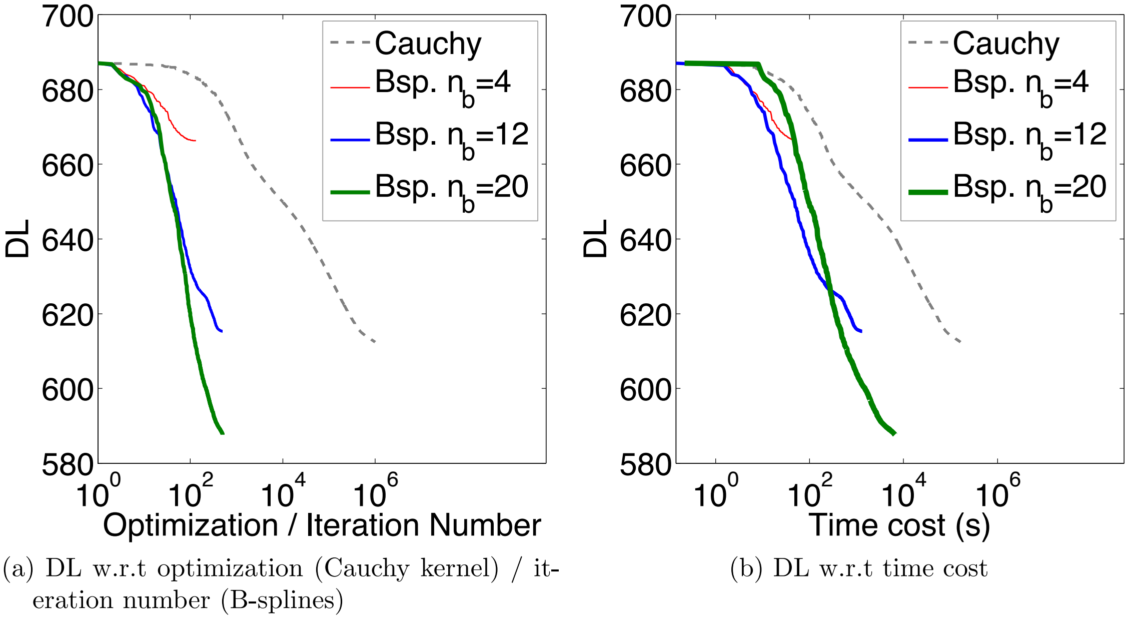
where σ is the sample standard deviation of $E_G(n_m)$

A third useful quantitative measure is the compactness error defined by

$$E_C(n_m) = \sum_{m=1}^{n_m} \sqrt{\frac{2}{n_S} \lambda_m} \tag{2.27}$$

It can be seen that the optimized statistical model's quality has significantly improved in that the optimized correspondence with SSM leads to smaller error than that of the initial correspondence for all three measures.

Time cost comparison. The time efficiency between the concatenation of Cauchy kernels and B-spline based reparameterization at coefficient resolution of $n_b = 4, 12, 20$ are compared in Figure 2.29, where the DL history is plotted with respect to both optimization/iteration number and time cost in seconds. In order for DL to reach the level of 620, Cauchy kernel concatenation takes 10^5 iterations and 1.6×10^5 s, and direct reparameterization with B-spline at resolution $n_b = 12$ only needs 502 iterations and 1.3×10^3 s. Therefore the B-spline based direct reparameterization is around 100 times more efficient than the concatenation of Cauchy kernels based reparameterization.



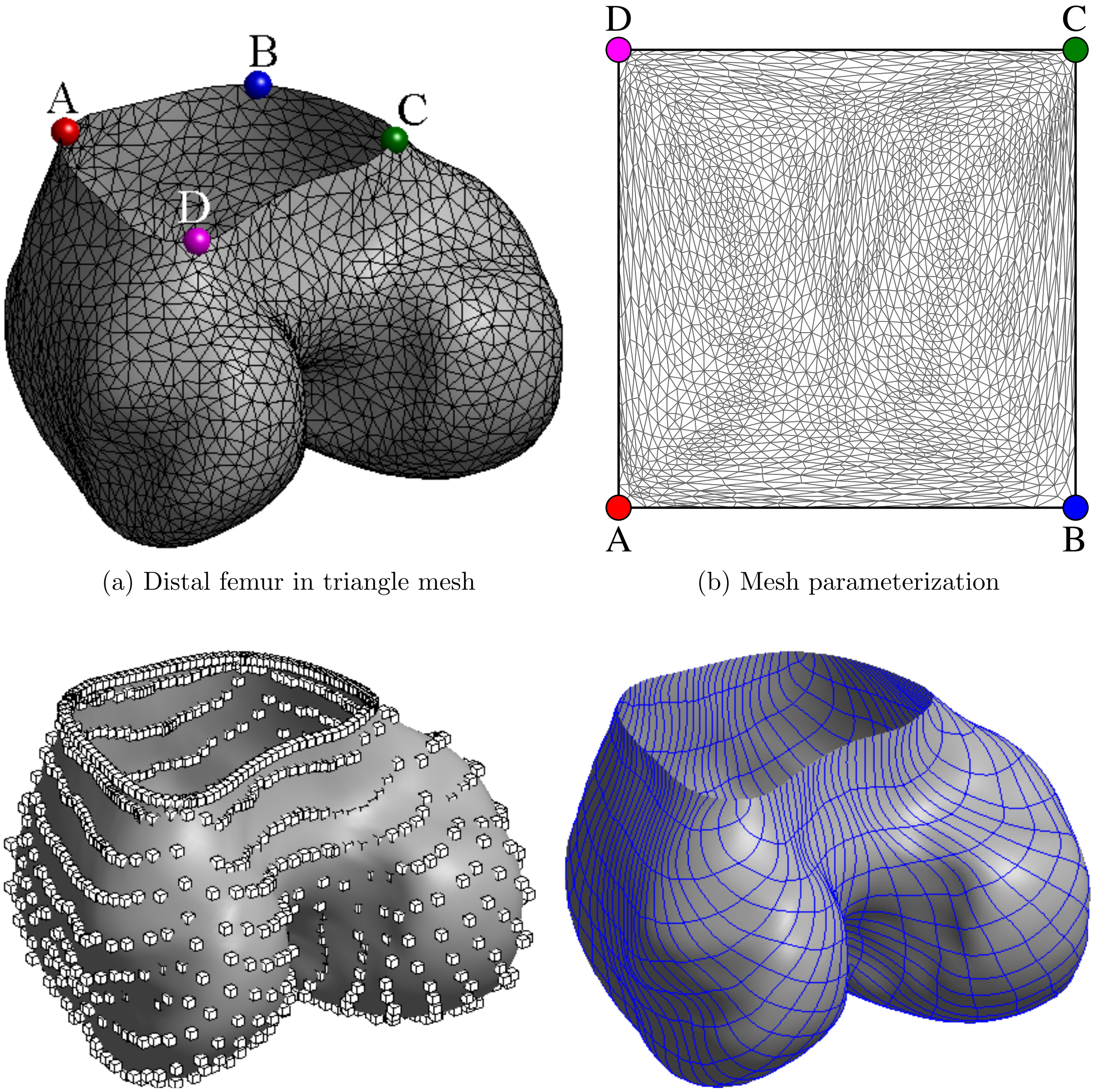
(a) DL w.r.t optimization (Cauchy kernel) / iteration number (B-splines)

(b) DL w.r.t time cost

Figure 2.29. Time cost comparison: concatenation of Cauchy kernels vs. B-spline based reparameterization for correspondence optimization

2.3.5 3D real data: Distal femur.

2.3.5.1 Training set pre-processing. The second real data set is a group of 34 human femoral bones. In this study, we focus on the shape variation of the distal femur portion, which is separated from the whole femur with a manual planar cut for each shape instance. Figure 2.30(a) shows the distal femur shape instance of Shape 1 as an example. For our pre-processing of the mesh models into the B-spline surfaces, we obtain the mesh parameterization with a method proposed in [57] that map an open mesh into a rectangle domain. To enhance the mesh parameterization and to avoid unevenness of the triangles in the parametric domain, a stretch measure suggested in [58] is then minimized and the obtained mesh parameterization is shown in Figure 2.30(b).



(a) Distal femur in triangle mesh

(b) Mesh parameterization

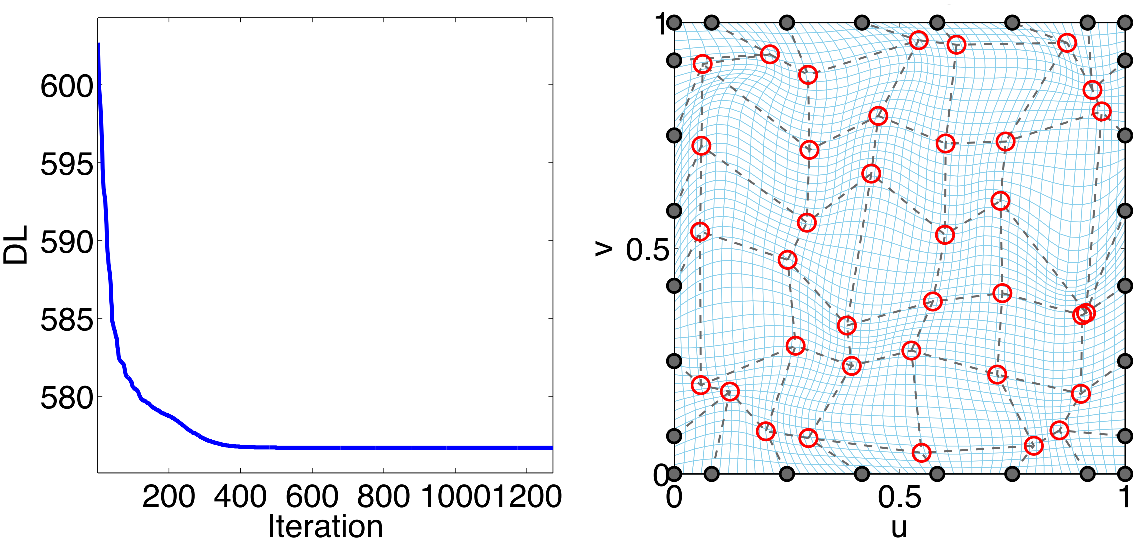
(c) Bi-quadratic B-spline with 30 × 30 control points (white cubes)

(d) B-spline and knot curves

Figure 2.30. Pre-processing the distal femoral bone

The 4 corners denoted by A, B, C, D for each mesh are manually chosen and considered to be in good correspondence across the training set, so that only the interior correspondence needs to be optimized. With the parameterization of each mesh vertex, a bi-quadratic B-spline surface of 30 × 30 control points is then fit on each shape. The resulting B-spline representation of the mesh shown in Figure 2.30(a) is now shown in Figure 2.30(c) and (d). This B-spline represented shape provides a differentiable shape representation for correspondence manipulation. Similar to the hand example, a one-time alignment is performed only once before the correspondence optimization. It is done by aligning the triangle meshes with the Iterative Closest Point algorithm.

2.3.5.2 Optimization via reparameterization B-spline. In the optimization,

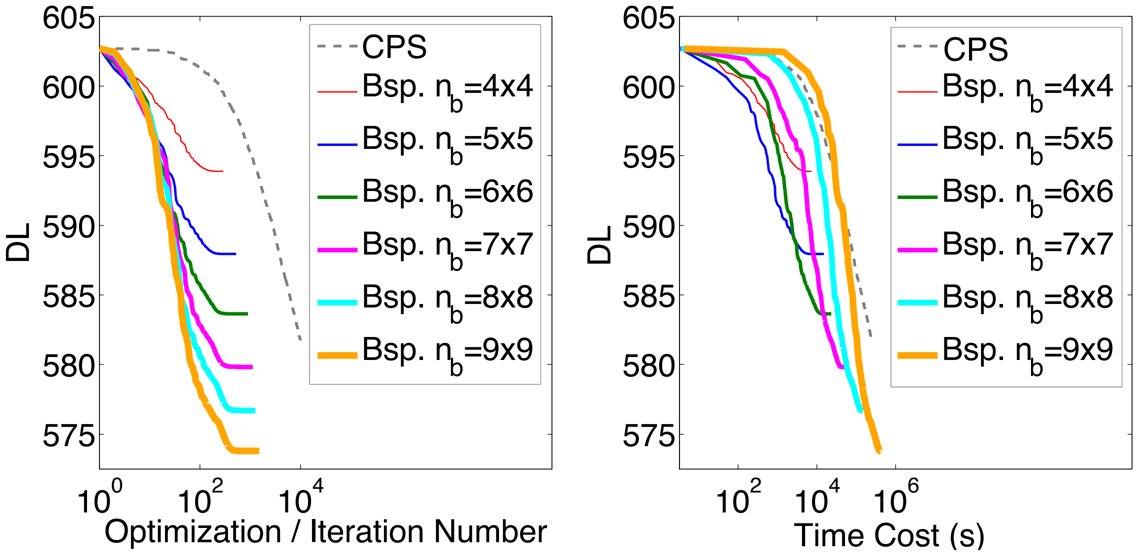


(a) DL history

(b) Parametric grid under optimized $\mathbf{D}_{34}(\mathbf{u})$

Figure 2.31. Optimization DL history and result of deformed grid of Shape 34 under the optimized reparameterization function with B-spline coefficients resolution of 8 × 8.

Shape 1 is selected as reference. Totally 51 × 51 landmarks are sampled uniformly in the parametric domain of each shape. When the reparameterization B-spline coefficients resolution is chosen to be a 8 × 8 grid, the objective DL history is shown in Figure 2.31(a), and the deformed parameterization grid under the optimized reparameterization function for Shape 34 is visualized in Figure 2.31(b). The quantitative evaluation of the optimized SSM in generalization, specificity error and compactness are also performed and sizable improvements similar to Figure 2.28 are observed.



(a) DL w.r.t optimization (CPS) / iteration number (B-splines)

(b) DL w.r.t time cost

Figure 2.32. Time cost comparison: concatenations of CPS warps vs. direct B-spline based reparameterization for correspondence optimization for 34 shapes.

Figure 2.32 shows the time cost comparison between concatenations of Clamped Plate Spline warps (optimization stopped till the 10^4-th iteration) vs. direct B-spline based reparameterization (B-spline coefficients grid from 4×4 to 9×9) for correspondence optimization for this femur data set. It can be seen that the direct approach with 6×6 reaches the same level of DL around 585 at approximately two orders of magnitude more efficient rate. With the further increase of the resolution (i.e. the number of B-spline coefficients) of the B-spline reparameterization functions, the DL can be reduced further than that of concatenation of CPSs, although the computational efficiency advantage becomes gradually less noticeable.

CHAPTER 3

COVARIANCE MATRIX OF A SHAPE POPULATION: A TALE ON SPLINE SETTING

3.1 Covariance matrix of spline curves and surfaces

For the above continuous formulations of the covariance matrix, (1.23) and (1.25), we show they can be computed efficiently and accurately with Bézier/B-spline based shape representation, either in closed-form or with quadrature methods.

3.1.1 Continuous formulation I with analytical integral. With training set shapes in the Bézier form, the integrand in continuous formulation I (1.23) 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 (1.23).

3.1.1.1 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

$$\mathbf{S}_i(u) = \sum_{j=0}^{p} B_j^p(u)\mathbf{P}_j^{(i)}, \qquad u \in [0,1] \tag{3.1}$$

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

$$\bar{\mathbf{S}}(u) = \frac{1}{n_S}\sum_{i=1}^{n_S} \mathbf{S}_i(u) = \sum_{j=0}^{p} B_j^p(u)\bar{\mathbf{P}}_j, \tag{3.2}$$

which is still a degree p Bézier curve with control points

$$\bar{\mathbf{P}}_j = \frac{1}{n_S}\sum_{i=1}^{n_S} \mathbf{P}_j^{(i)}. \tag{3.3}$$

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

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

The covariance matrix as formulated in (1.23) 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

$$Q^{(i_1,i_2)}_j = \sum_{l=\max(0,j-p)}^{\min(j,p)} \frac{\binom{p}{l}\binom{p}{j-l}}{\binom{2p}{j}}\, \widehat{\mathbf{P}}^{(i_1)T}_{l}\widehat{\mathbf{P}}^{(i_2)}_{j-l}. \tag{3.4}$$

In the above equation, we utilize the fact the product of two Bernstein polynomials of degree p and q is a higher order Bernstein polynomial of degree $(p+q)$ as proved in [59], 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 Bernstein polynomial below [59]

$$\int_0^1 \sum_{j=0}^{p} B^p_j(u) Q_j \,\mathrm{d}u = \frac{\sum_{j=0}^{p} Q_j}{p+1}, \tag{3.5}$$

the covariance matrix expression reduces to

$$C^I_{i_1 i_2} = \sum_{j=0}^{2p} \sum_{l=\max(0,j-p)}^{\min(j,p)} \frac{\binom{p}{l}\binom{p}{j-l}\widehat{\mathbf{P}}^{(i_1)T}_{l}\widehat{\mathbf{P}}^{(i_2)}_{j-l}}{\binom{2p}{j}(n_S-1)(2p+1)}. \tag{3.6}$$

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.

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

$$\mathbf{S}_i(\mathbf{u}) = \sum_{j=0}^{p}\sum_{k=0}^{q} B_j^p(u) B_k^q(v) \mathbf{P}_{j,k}^{(i)}, \qquad u \times v \in [0,1]^2, \tag{3.7}$$

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

$$\begin{aligned} C_{i_1 i_2}^I = & \sum_{j=0}^{2p}\sum_{k=0}^{2q}\sum_{l=\max(0,j-p)}^{\min(j,p)}\sum_{m=\max(0,k-q)}^{\min(k,q)} \\ & \frac{\binom{p}{l}\binom{p}{j-l}\binom{q}{m}\binom{q}{k-m}\widehat{\mathbf{P}}_{l,m}^{(i_1)T}\widehat{\mathbf{P}}_{j-l,k-m}^{(i_2)}}{\binom{2p}{j}\binom{2q}{k}(n_S-1)(2p+1)(2q+1)}. \end{aligned} \tag{3.8}$$

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 $\bar{\mathbf{P}}_{j,k} = \frac{1}{n_S}\sum_{i=1}^{n_S} \mathbf{P}_{j,k}^{(i)}$.

Since a B-spline curve (surface) is simply a piecewise collection of Bézier curves (surfaces), the piecewise summation of (3.6) and (3.8) gives the analytical integrals for for shapes represented in B-splines.

3.1.2 Continuous formulation II with analytical integrand.

Due to the Jacobian involved in continuous formulation II (1.25), even for shapes that are parameterized by Béziers or B-splines, the analytical integration is difficult to obtain in general. However, analytical form of the integrand can still be obtained, which would facilitate the quadrature based numerical integration.

3.1.2.1 Bézier curves. If all shapes are represented in Bézier curves of degree p with the definition presented in (3.1). The Jacobian is degree $(p-1)$ Bézier defined by

$$\mathbf{J}(u) = \sum_{k=0}^{p-1} B_k^{p-1}(u) \widetilde{P}_k, \tag{3.9}$$

where the new control points are

$$\widetilde{P}_k = p(\bar{P}_{k+1} - \bar{P}_k). \tag{3.10}$$

Therefore the covariance matrix in this context reduces to

$$C^{II}_{i_1 i_2} = \frac{\displaystyle\int_0^1 \sum_{j=0}^{2p} \sum_{k=0}^{p-1} B_j^{2p}(u) B_k^{p-1}(u) Q_j^{(i_1,i_2)} \widetilde{P}_k \, \mathrm{d}u}{\displaystyle (n_S - 1) \int_0^1 \sum_{k=0}^{p-1} B_k^{p-1}(u) \widetilde{P}_k \, \mathrm{d}u}, \tag{3.11}$$

where $Q_j^{(i_1,i_2)}$ is defined earlier in (3.4).

3.1.2.2 Bézier surfaces and B-spline curves/surfaces. Similarly 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.

3.1.3 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 (1.21). Such approximated continuous forms would reveal the link between continuous form and the usual discrete form. We examine two integration schemes below: mid-point rule and Gaussian quadrature.

3.1.3.1 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

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

where the mid-point parameter is $\xi_j = (\widetilde{u}_j + \widetilde{u}_{j+1})/2$. Due to equal intervals and recall 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 (1.14), and the above could be written as

$$\begin{aligned}\widetilde{C}_{i_1 i_2}^{I,Mid} &= \frac{1}{n_S - 1} \sum_{j=1}^{n_P} \left[\mathbf{x}_{i_1}^{(j)} - \bar{\mathbf{x}}\right]^T \left[\mathbf{x}_{i_2}^{(j)} - \bar{\mathbf{x}}\right] \frac{1}{n_P} \\ &= \frac{1}{(n_S - 1) n_P} \left(\mathbf{X}_{i_1} - \bar{\mathbf{X}}\right)^T \left(\mathbf{X}_{i_2} - \bar{\mathbf{X}}\right).\end{aligned} \tag{3.12}$$

This shows the discrete formulation (1.15) or (1.20a) is equivalent to the approximated continuous formulation I (3.12)

$$\widetilde{C}_{i_1 i_2}^{I,Mid} = \frac{1}{n_P} \widetilde{D}_{i_1 i_2}. \tag{3.13}$$

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

Integration by Gaussian quadrature. Besides the mid-point rule to evaluate the integral in (1.23), another common form of numerical integration is through Gaussian quadrature. Since the integrand in (1.23) is just degree $2p$ polynomials for curves, 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.

3.1.3.2 Continuous formulation II.

Integration by mid-point rule. The above numerical integration approaches can be similarly used to obtain the approximate forms for continuous formulation II. The covariance matrix of the continuous formulation II in (1.25) for the curve case can be approximated by

$$\widetilde{C}_{i_1 i_2}^{II,Mid} = \frac{\sum_{j=1}^{n_P} \left[\mathbf{S}_{i_1}(\xi_j) - \bar{\mathbf{S}}(\xi_j)\right]^T \left[\mathbf{S}_{i_2}(\xi_j) - \bar{\mathbf{S}}(\xi_j)\right] \Delta L(\xi_j)}{(n_S - 1) \sum_{j=1}^{n_P} \Delta L(\xi_j)}, \tag{3.14}$$

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

$$\Delta L(\xi_j) = \begin{cases} \frac{\left|\bar{\mathbf{S}}(\xi_{j+1})-\bar{\mathbf{S}}(\xi_j)\right|}{2} & j=1 \\ \frac{\left|\bar{\mathbf{S}}(\xi_j)-\bar{\mathbf{S}}(\xi_{j-1})\right|+\left|\bar{\mathbf{S}}(\xi_{j+1})-\bar{\mathbf{S}}(\xi_j)\right|}{2} & 1<j<n_P \\ \frac{\left|\bar{\mathbf{S}}(\xi_j)-\bar{\mathbf{S}}(\xi_{j-1})\right|}{2} & j=n_P \end{cases} \tag{3.15}$$

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

$$\widetilde{C}^{II,Mid}_{i_1 i_2} = \frac{\sum_{j=1}^{n_{Pu}} \sum_{k=1}^{n_{Pv}} \widehat{\mathbf{S}}_{i_1}(\xi_j,\eta_k)\widehat{\mathbf{S}}_{i_1}(\xi_j,\eta_k)\Delta A(\xi_j,\eta_k)}{(n_S-1)\sum_{j=1}^{n_{Pu}} \sum_{k=1}^{n_{Pv}} \Delta A(\xi_j,\eta_k)}, \tag{3.16}$$

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 shown that the approximate continuous formulation II (3.16) for the surface case is equivalent to the discrete form (1.15) and (1.20a) 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 as compared to that of continuous formulation I is the added the Jacobian term and its normalizer. Since the Jacobian term $|\mathbf{J}|$ is not a polynomial over the parameter domain due to the square root norm, analytical form cannot be derived for the formulation II ; however, analytical integrand can be obtained in Gauss integration.

3.2 Covariance matrix under reparameterization

3.2.1 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 B-spline representation $\mathbf{S}(\mathbf{u})$ of the shapes. In order 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.

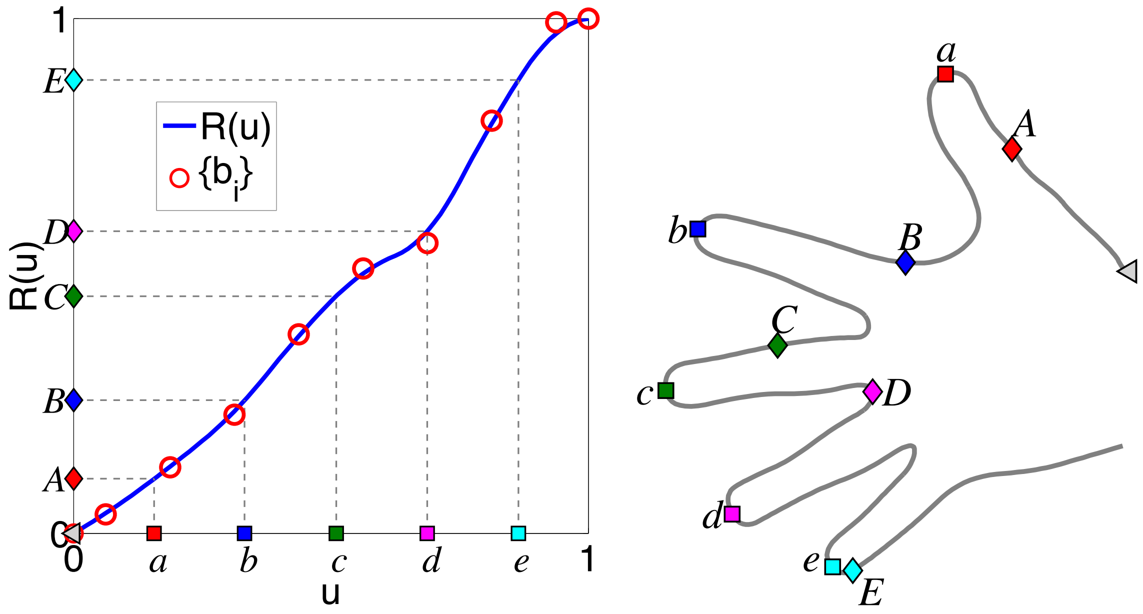
3.2.1.1 Reparameterization of curves. Reparameterization function in the curve case $\mathbf{R}(\mathbf{u})$ could be directly represented by a degree d B-spline function as

$$R(u) = \sum_{i=0}^{n_B} N_i^d(u) b_i, \qquad 0 \leqslant u \leqslant 1, \tag{3.17}$$

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 $R(1) = 1$.

Figure 3.1(a) shows 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 could be observed by five points, which moved 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 also relocated to somewhere else as seen in a B-spline represented finger tip in Figure 3.1(b).

In order to have a valid reparameterization that is free of self-intersection, it is required to enforce the bijectivity constraint for diffeomorphic reparameterization



(a) Reparametrization 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 3.1. Direct reparameterization of a B-spline curve

by setting $\mathrm{d}\,R(u)/\mathrm{d}\,u > 0$. Since $R(u)$ 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 [34], we thus have the following explicit constraint for ensuring diffeomorphic reparameterization of curves

$$b_i - b_{i+1} < 0, \qquad i = 0, 1, ..., n_B - 1. \tag{3.18}$$

3.2.1.2 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

$$\mathbf{R}(\mathbf{u}) = \sum_{i=0}^{n_{B_u}} \sum_{j=0}^{n_{B_v}} N_i^d(u) N_j^e(v) \mathbf{b}_{i,j}, \quad 0 \leqslant u, v \leqslant 1, \tag{3.19}$$

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 respectively defined on two sets of non-decreasing knot vector $U = \{\bar{u}_0, \bar{u}_1, ..., \bar{u}_{n_{B_u}+d+1}\}$ and

$V = \{\bar{v}_0, \bar{v}_1, ..., \bar{v}_{n_{B_v}+e+1}\}$.

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 and 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.

$$\mathcal{J}(\mathbf{u}) = \begin{vmatrix} \frac{\partial R_u(u,v)}{\partial u} & \frac{\partial R_u(u,v)}{\partial v} \\ \\ \frac{\partial R_v(u,v)}{\partial u} & \frac{\partial R_v(u,v)}{\partial v} \end{vmatrix} > 0, \quad \forall (u,v) \in [0,1]. \tag{3.20}$$

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.

3.2.2 Incorporation of reparameterization into the covariance matrix.

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

3.2.2.1 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 (1.23) becomes

$$C^I_{i_1 i_2} \doteq \frac{1}{n_S - 1} \int_{\mathcal{U}} \widehat{\mathbf{S}}_{i_1}[\mathbf{R}_{i_1}(\mathbf{u})]^T \widehat{\mathbf{S}}_{i_2}[\mathbf{R}_{i_2}(\mathbf{u})] \, \mathrm{d}\mathbf{u}, \tag{3.21}$$

where

$$\widehat{\mathbf{S}}_i[\mathbf{R}_i(\mathbf{u})] = \mathbf{S}_i\left[\mathbf{R}_i(\mathbf{u})\right] - \bar{\mathbf{S}}(\mathbf{R}(\mathbf{u})), \tag{3.22}$$

and the *continuous mean shape* (1.24) becomes

$$\bar{\mathbf{S}}(\mathbf{R}(\mathbf{u})) \doteq \frac{1}{n_S}\sum_{i=1}^{n_S}\mathbf{S}_i[\mathbf{R}_i(\mathbf{u})]. \tag{3.23}$$

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 just 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 Bernstein 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

$$\mathbf{S}_{i;s}(u) = \sum_{j_S=0}^{p} B^p_{j_S}(\hat{u})\mathbf{P}^{(i;s)}_{j_S}, \tag{3.24}$$

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

$$R_{i;s}(u) = \sum_{j_R=0}^{d} B^d_{j_R}(\hat{u})\, b^{(i;s)}_{j_R}, \tag{3.25}$$

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 [59]

$$\mathbf{S}_{i;s}[R_{i;s}(u)] = \sum_{j=0}^{pd} (1-\hat{u})^{pd-j} \hat{u}^j \; \widetilde{\mathbf{Y}}_j^{(i;s)}, \tag{3.26}$$

where $\widetilde{\mathbf{Y}}_j$ is a sum of the scaled Bernstein coefficients. Releasing the binomial coefficients in (3.26) restores the Bernstein basis

$$\mathbf{S}_{i;s}[R_{i;s}(u)] = \sum_{j=0}^{pd} B_j^{pd}(\hat{u}) \; \mathbf{Y}_j^{(i;s)}, \text{s.t. } \mathbf{Y}_j^{(i;s)} = \widetilde{\mathbf{Y}}_j^{(i;s)} \Big/ \binom{pd}{j} \tag{3.27}$$

The mean shape is still a degree-*pd* Bézier curve

$$\bar{\mathbf{S}}_s(u) = \sum_{j=0}^{pd} B_j^{pd}(\hat{u}) \bar{\mathbf{Y}}_j^{(s)}, \text{where: } \bar{\mathbf{Y}}_j^{(s)} = \frac{1}{n_S} \sum_{i=1}^{n_S} \mathbf{Y}_j^{(i;s)}, \tag{3.28}$$

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

$$\hat{\mathbf{S}}_{i;s}[R_{i;s}(u)] = \mathbf{S}_{i;s}[R_{i;s}(u)] - \bar{\mathbf{S}}_s(u) = \sum_{j=0}^{pd} B_j^{pd}(\hat{u}) \mathbf{Z}_j^{(i;s)}, \tag{3.29}$$

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

$$\begin{aligned} C^I_{i_1 i_2} &= \int_0^1 \frac{1}{n_S - 1} \hat{\mathbf{S}}_{i_1;s}[R_{i_1;s}(u)]^T \; \hat{\mathbf{S}}_{i_2;s}[R_{i_2;s}(u)] \, \mathrm{d}u \\ &= \frac{\sum_{s=1}^{n_K} \left(\tilde{\xi}_{p+s} - \tilde{\xi}_{p+s-1} \right) \sum_{j=0}^{2pd} W_j^{(i_1,i_2;s)}}{(n_S - 1)(2pd+1)}, \end{aligned} \tag{3.30}$$

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

B-spline curves under B-spline reparameterization. B-spline curves generally have more than one knot span, an inverse map has to be computed to locate those knot spans after reparameterization since 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

$$R(\psi_k) = \xi_k \qquad k = 0, 1, ..., n+p+1. \tag{3.31}$$

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 (3.26) holds. The covariance matrix can be analytically computed by (3.30) 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.

To sum up, we can evaluate the covariance matrix in continuous formulation I analytically by strictly following the analytical formulas listed in Table 3.1.

Table 3.1. Analytical formulas for continuous formulation I with/without B-spline reparameterization

Training set shape type	Without reparam.	With reparam.
Bézier curves	(3.6)	(3.30)
Bézier surfaces	(3.8)	—
B-spline curves	(3.6)	(3.30)
B-spline surfaces	(3.8)	—

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 shown in the previous section.

Table 3.2 gives the minimum number of quadrature points required for exact recovery of the covariance matrix of continuous formulation I. The first four rows are for splines represented shapes without reparameterization and the last four rows with reparameterization. Note, 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 3.2. 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	$p \times q$	$d \times e$	—

3.2.2.2 Continuous formulation II.

With reparameterization of the shapes, the basic form (1.25) of the covariance matrix in continuous formulation II becomes

$$C_{i_1 i_2}^{II} \doteq \frac{\displaystyle\int_{\mathcal{U}} \widehat{\mathbf{S}}_{i_1}[\mathbf{R}_{i_1}(\mathbf{u})]^T \widehat{\mathbf{S}}_{i_2}[\mathbf{R}_{i_2}(\mathbf{u})]\, |\mathbf{J}(\mathbf{R}(\mathbf{u}))|\, \mathrm{d}\mathbf{u}}{\displaystyle (n_S - 1) \int_{\mathcal{U}} |\mathbf{J}(\mathbf{R}(\mathbf{u}))|\, \mathrm{d}\mathbf{u}} \tag{3.32}$$

where the *continuous mean shape* (1.24) becomes (3.23), $\mathbf{J}(\mathbf{R}(\mathbf{u}))$ is computed from the mean shape, and $\widehat{\mathbf{S}}_i[\mathbf{R}_i(\mathbf{u})]$ is defined as in (3.22).

The analytical integrand can be similarly achieved with the procedure described in Section 3.1.2.1. This analytical integrand can then be used in the mid-point or Gauss quadrature method for integrating entry in the covariance matrix.

3.3 Shape correspondence optimization via reparameterization

With the B-spline representation of reparameterization functions $\mathbf{R}(\mathbf{u})$ and the diffeomorphic conditions (3.18) and (3.20) presented in Section 3.2.1.1 and 3.2.1.2, we thus again have the optimization formulation in (2.12d) , where $\mathbf{b}$ 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 [22], which is a function of eigenvalues computed from (2.12b) and (2.12c). 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 (2.12d) represents the diffeomorphic conditions, i.e. (3.18) for curves and (3.20) for surfaces, each of which is a function of optimization variables $\mathbf{b}$.

3.4 Numerical examples

In this section, we compare numerical results of computed covariance matrices from two continuous formulations under different discretization resolutions. In order to compare the analytical form of the covariance matrix (1.23) from continuous formulations and their approximations through mid-point or Gauss quadrature and their convergence, we compare the the covariance matrix norm and its largest eigenvalue. The matrix norm used is the Frobenius norm of a $m \times n$ matrix $\mathbf{C} = \{C_{ij}\}$ $(i = 1, ..., m; j = 1, ..., n)$

$$|\mathbf{C}| = \sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n} C_{ij}^2}. \tag{3.33}$$

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

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

In order 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 shown in Figure 3.2(a). These four quarter circle shapes can be represented with the following parametric form

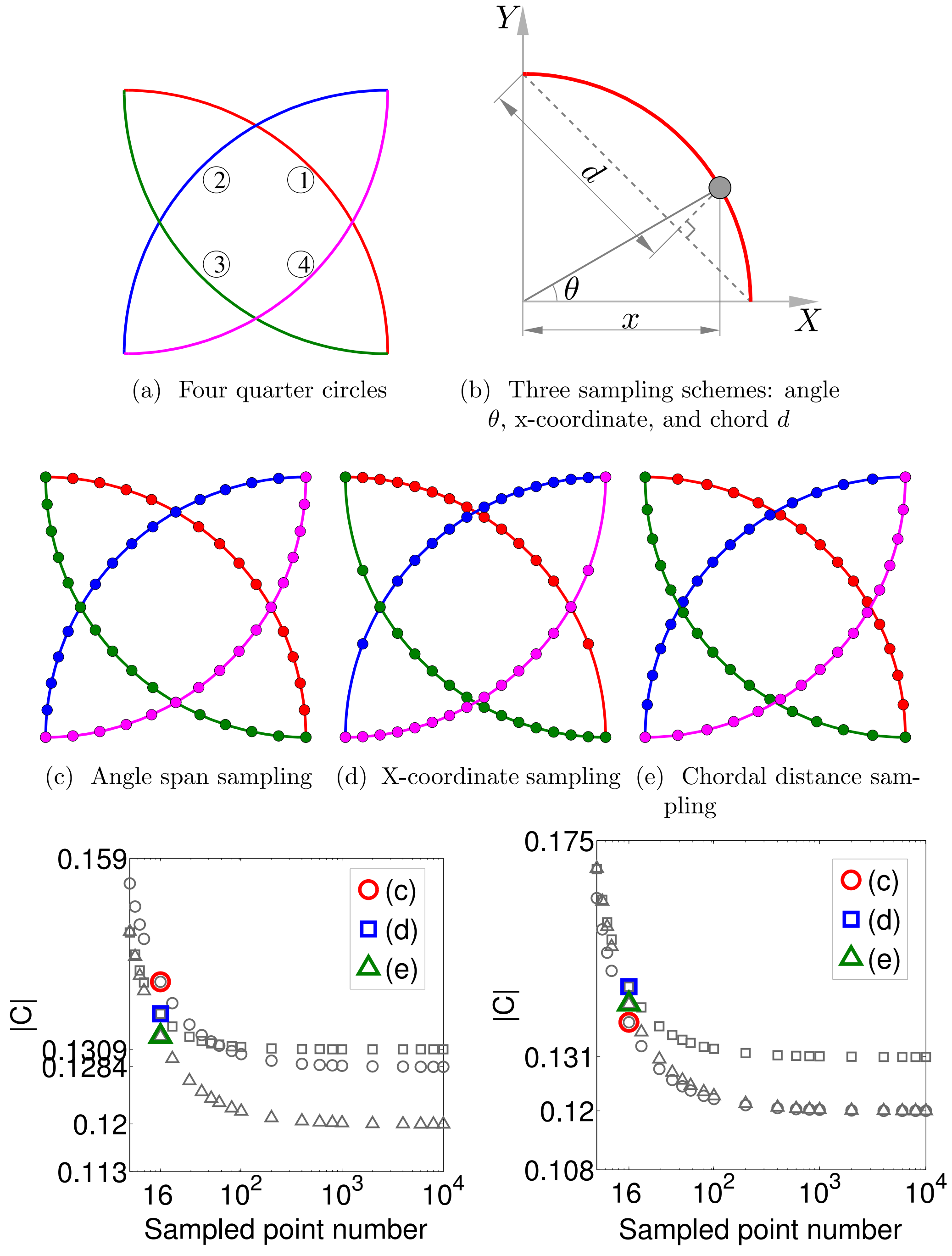
$$\begin{aligned}
\mathbf{S}_1(u) &= \left[\sin\left(\frac{\pi}{2}u\right), \cos\left(\frac{\pi}{2}u\right)\right] \\
\mathbf{S}_2(u) &= \left[1 - \cos\left(\frac{\pi}{2}u\right), \sin\left(\frac{\pi}{2}u\right)\right] \\
\mathbf{S}_3(u) &= \left[1 - \cos\left(\frac{\pi}{2}u\right), 1 - \sin\left(\frac{\pi}{2}u\right)\right] \\
\mathbf{S}_4(u) &= \left[\sin\left(\frac{\pi}{2}u\right), 1 - \cos\left(\frac{\pi}{2}u\right)\right]
\end{aligned} \tag{3.34}$$

where $u \in [0, 1]$. With such explicit, continuous representations of arcs, the covariance matricies of continuous formulation I (1.23) and formulation II (1.25) can be directly computed, without resort to discrete sampling or B-spline fitting. The resulting matrix norms are respectively 0.1284 for formulation I and 0.1200 for formulation II. We then compute the two forms of covariance matrices from 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 3.2(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 since the underling shapes are circular.

Figure 3.2(c), (d) and (e) respectively show the sampled points based on the three sampling schemes where 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 (1.20a) is shown in Figure 3.2(f), where the x-axis indicates the number of sampled points that ranges from 6 to 10^4. The norm of another weighted discrete covariance matrix (3.14) with substituting sampled points $\mathbf{x}_i^{(j)}$ for $\mathbf{S}_i(\xi_j)$ is shown in Figure 3.2(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 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 shown in Figure 3.2(f). When using weighted points in the covariance matrix; they converge to 0.1200, 0.1310 and 0.1200 as shown in Figure 3.2(g). Among the three point sampling schemes, only angle span based sampling has a converged value consistent with reference value 0.1284 since it reflects the arc-length parameterization. This example clearly demonstrates that, for the covariance matrix from discrete points based representation of shapes, it takes a large number of data points 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 [32], knot determination where a set of knot parameters for B-spline shapes need to be determined, data parameterization where for each data point Q_j a corresponding parameter $\bar{u}_j$ needs to be decided, and control point calculation where control points P_i are computed so that the error between the



(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

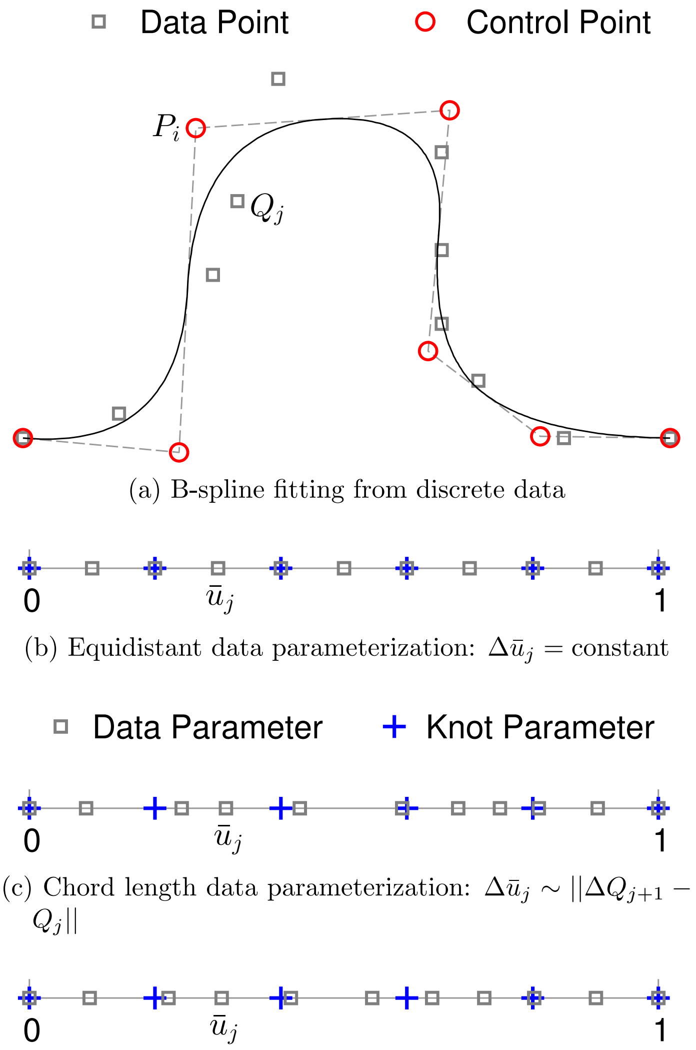
(g) Covariance matrix norm from weighted discrete points

Figure 3.2. Discrete sampling and corresponding covariance matrix norms. The sign (circle, square and triangle) in Figures 3.2(f) and 3.2(g) corresponds to respectively point sampling from angle span, x-coordinate and chord distance based schemes.

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 shown in Figure 3.3: a) equidistant $\Delta\bar{u}_j$ = constant; b) chord length $\Delta\bar{u}_j = ||\Delta Q_j||$; and c) the centripetal method $\Delta\bar{u}_i = \sqrt{||\Delta Q_{j+1} - Q_j||}$ [60] where Q_j represents the data point.



(a) B-spline fitting from discrete data

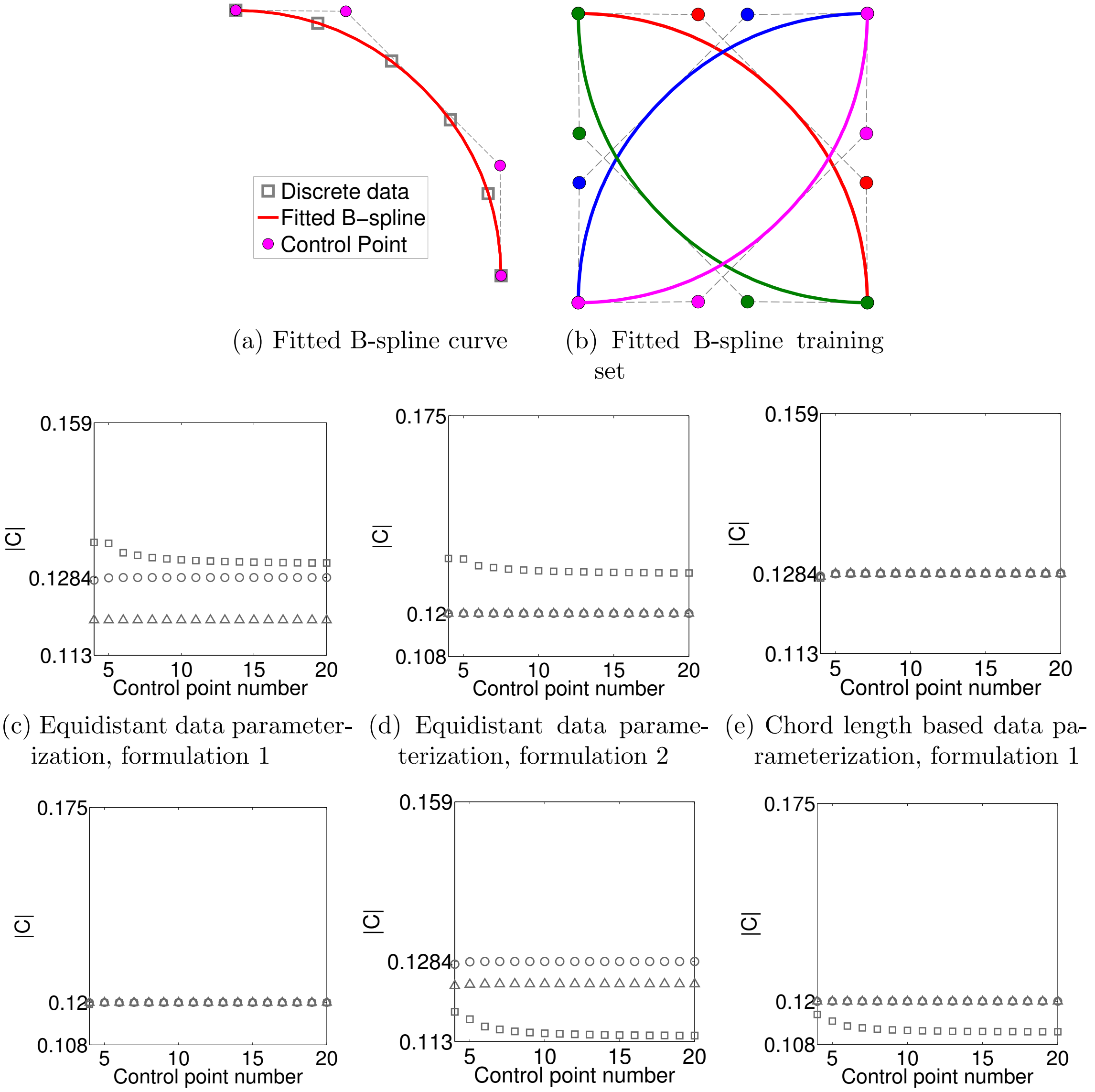
(b) Equidistant data parameterization: $\Delta\bar{u}_j$ = constant

(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 3.3. B-spline fitting with different methods of data parametrization.

Figure 3.4(a) shows an example of fitting n_P =6 sampled points with a quadratic



(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

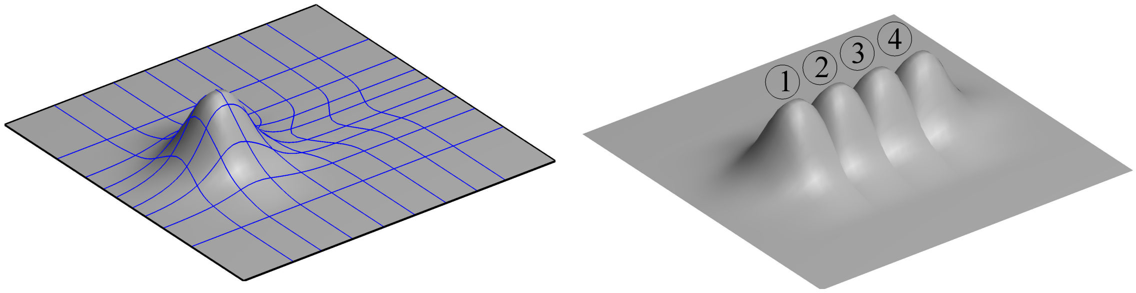
Figure 3.4. Covariance matrix norms from B-spline curves fitted with three methods of data parameterization. The sign (circle, square and triangle) corresponds to respectively angle span, x-coordinate and chord distance based point sampling scheme shown in Figures 3.2.

B-spline of $n_{CP} = 4$ control points, and Figure 3.4(b) is the fitted B-spine training set. In the convergence study of formulation I (computed by the analytical formula (3.6)) and formulation II computed by Gaussian quadrature approximation of (1.25) respectively, we choose $n_P = 100$ sampled points for B-spline curve fitting, and varying the number of control points n_{CP} from 4 to 20. The results of two continuous forms of covariance matrices computed with fitted B-splines curves are shown in the remainder of Figure 3.4 where three data parameterization methods for three sampling schemes are shown respectively in the last three rows in Figure 3.4. We can see from this figure that

- Even though covariance matrices from discrete points all need at least a thousand points to converge as shown in Figure 3.2, the convergence of the covariance matrix from B-splines only needs fewer than 10 control points as shown in Figure 3.4. 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 shown in Figure 3.4(e) and 3.4(f). This is because, when 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 representation 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.

3.4.2 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 shown in Figure 3.5(a) where the knot curves are shown in blue. The training set is composed of 4 such B-spline surfaces, which differ in horizontal position of the bump along the u-direction as seen in Figure 3.5(b).

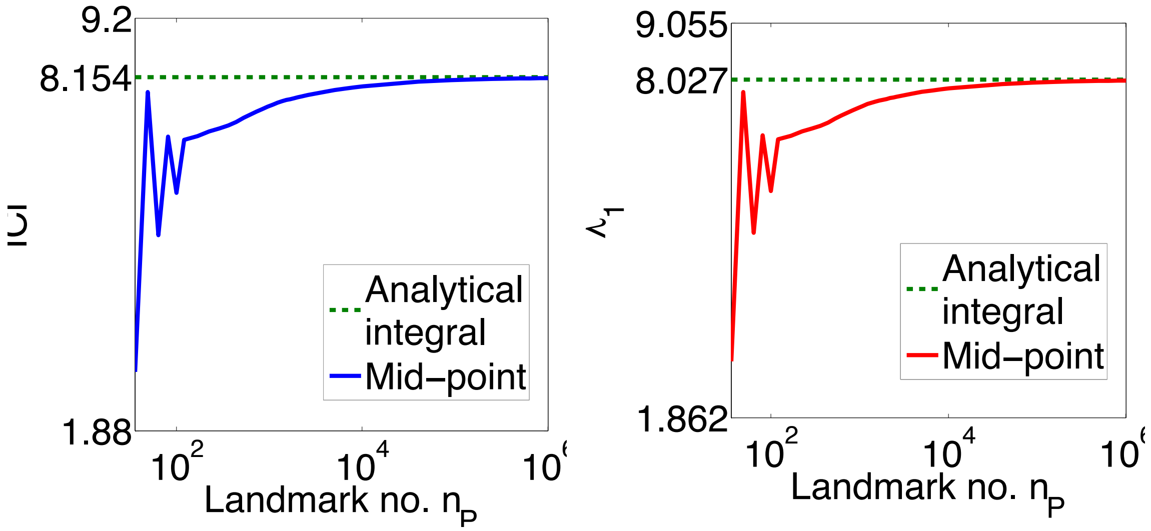


(a) Shape in bi-qaudratic B-spline surface

(b) 4 B-spline surfaces superimposed

Figure 3.5. Four B-spline represented plane-bumps superimposed. $n_S = 4; p = 2, q = 2$.

In the approximated continuous formulation I through mid-point, $n_P = n_{Pu} \times n_{Pv} = n_{Pu}^2$ 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 analytical covariance matrix is computed following derived equation (3.8). 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 3.6 that both the covariance matrix and the eigenvalues under the approximate formulation are approaching the analytical values as the number of landmarks tends 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.



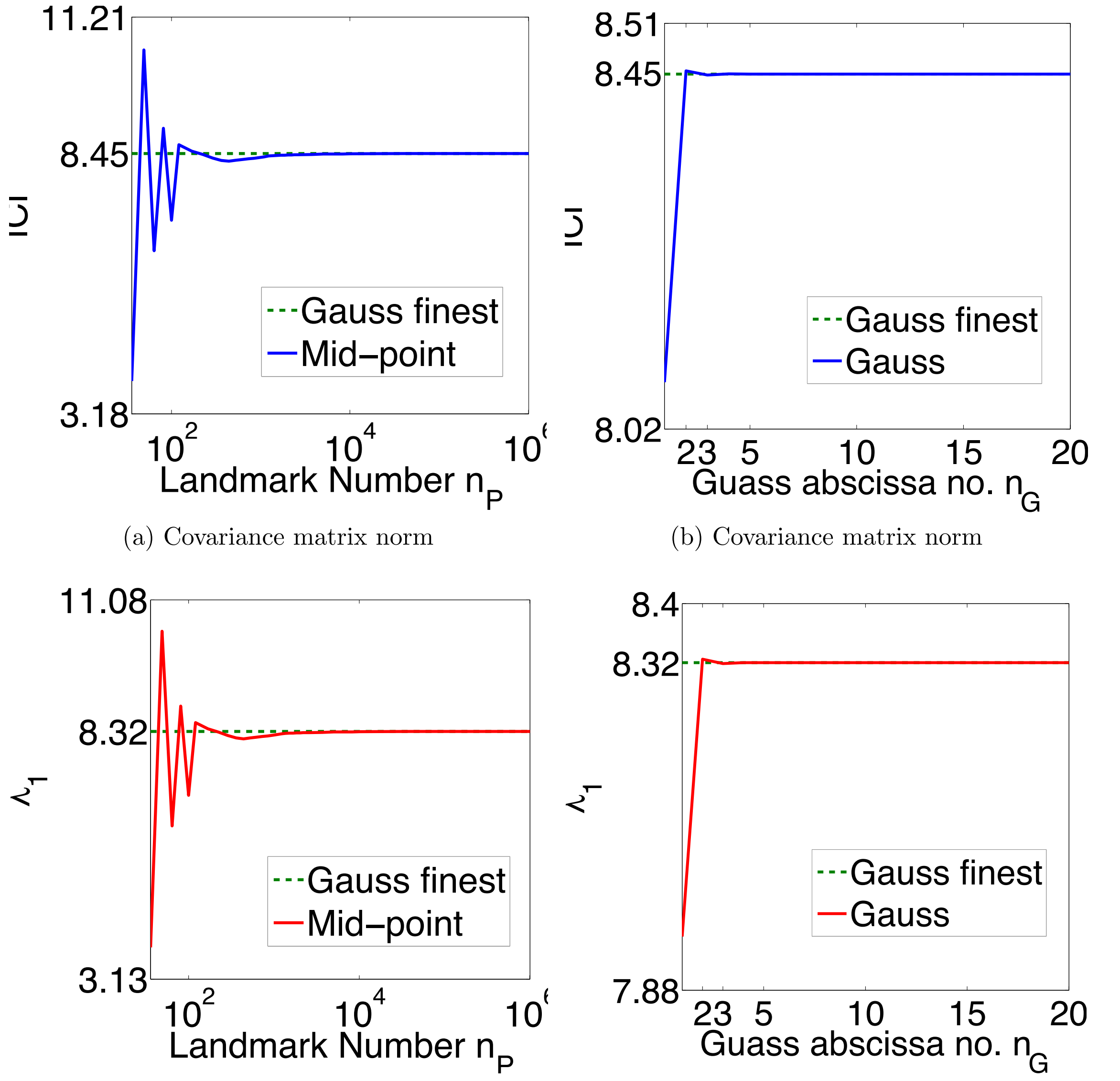
(a) Covariance matrix norm

(b) Largest eigenvalue

Figure 3.6. 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 Figure 3.5.

In the mid-point based continuous formulation II (3.16), $n_P = n_{Pu} \times n_{Pv} = n_{Pu}^2$ 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 3.7 shows that the covariance matrix and the eigenvalues computed with 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 3.6 and Figure 3.7, 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.



(a) Covariance matrix norm

(b) Covariance matrix norm

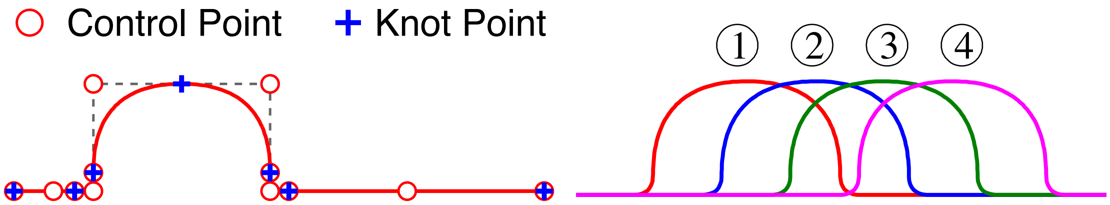
(c) Largest eigenvalue

(d) Largest eigenvalue

Figure 3.7. Convergence of the covariance matrix norm and eigenvalues with continuous formulation II and its approximations for the plane-bump training set shown in Figure 3.5.

3.4.3 Convergence of continuous formulations under reparameterization.

In this example, we show 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 shown in Figure 3.8(a). The training set then consists of 4 such B-spline curves, with only the horizontal position of the bump is different as displayed in Figure 3.8(b). This synthetic training set is just the open curve version of the benchmark box-bump as used in [19], with the bump feature further highlighted.



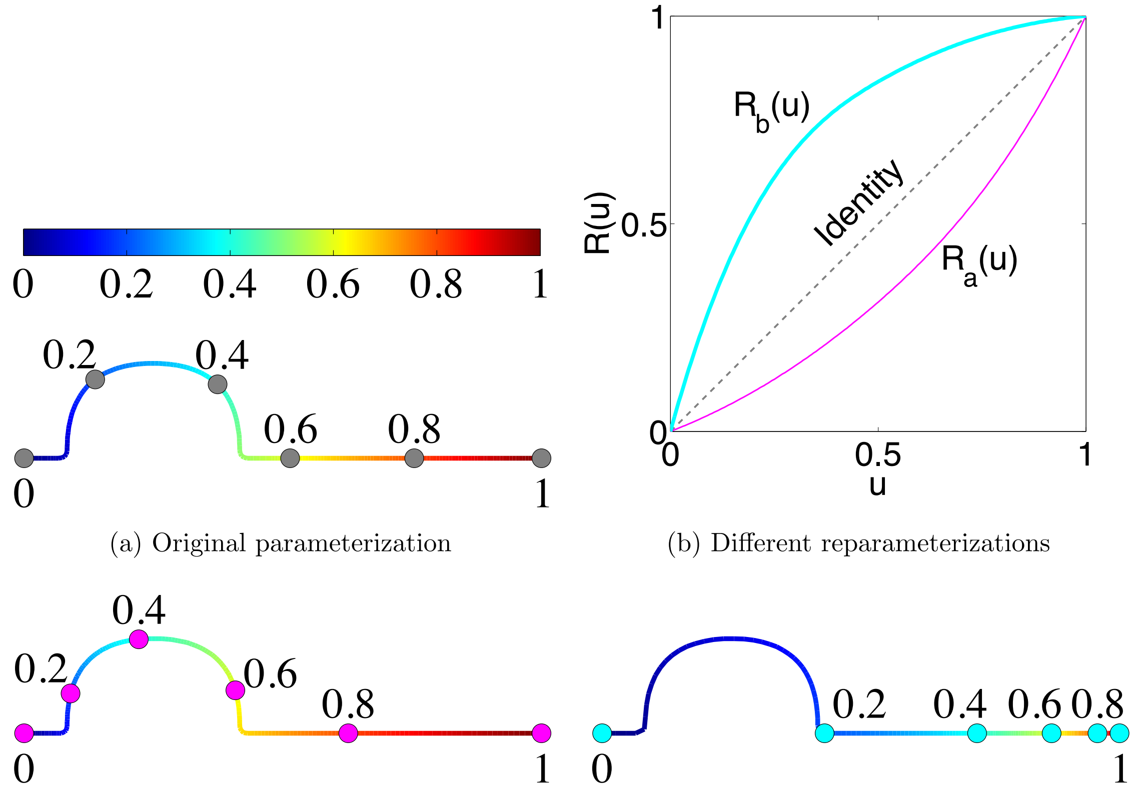
(a) Shape in quadratic B-spline of 12 control points

(b) 4 B-splines curves superimposed

Figure 3.8. Line-bump training set represented as B-spline curves. $n_S = 4; p = 2$.

The reparameterization functions are shown in Figure 3.9 where the color field of Figure 3.9(a) signifies the original parameterization of Shape 1. It is also shown as the identity function in Figure 3.9(b). Two reparameterization functions $R_a(u)$ and $R_b(u)$ in Figure 3.9(b) are used to generate different parameterizations in Figure 3.9(c) and Figure 3.9(d) respectively. $R_a(u)$ curved downward with a slight deviation from identity function makes the parameterization slightly squeezed toward $u = 0$; $R_b(u)$ curved upward with a large deviation from identity function makes the parameterization severely squeezed toward $u = 1$.

We applied the three reparameterization functions to all B-spline represented line-bumps. Figures 3.10 and Figure 3.11 respectively show the covariance matrix norm and the largest eigenvalues of the line-bump under the three different repa-



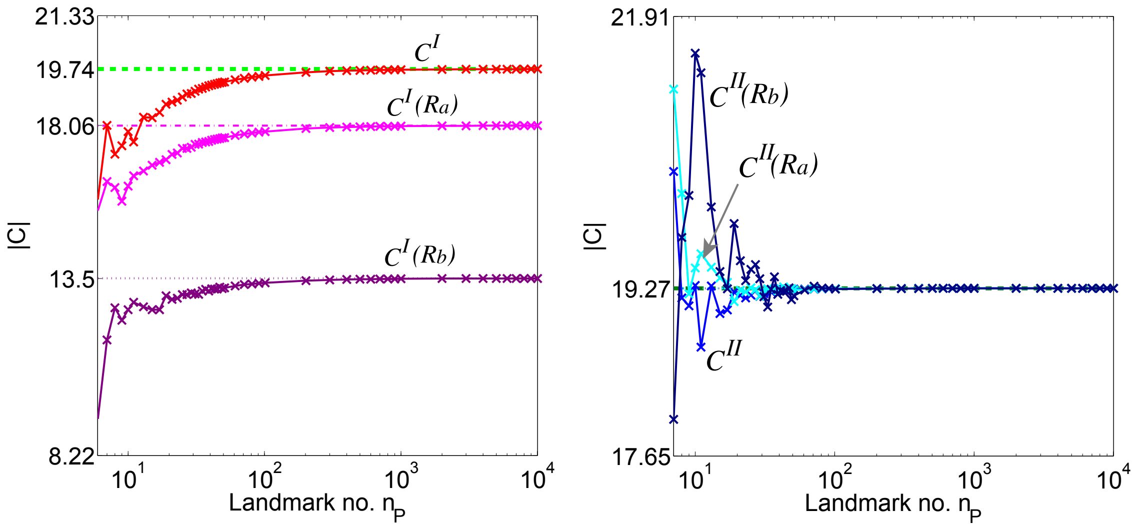
(a) Original parameterization

(b) Different reparameterizations

(c) Reparameterization $R_a(u)$

(d) Reparameterization $R_b(u)$

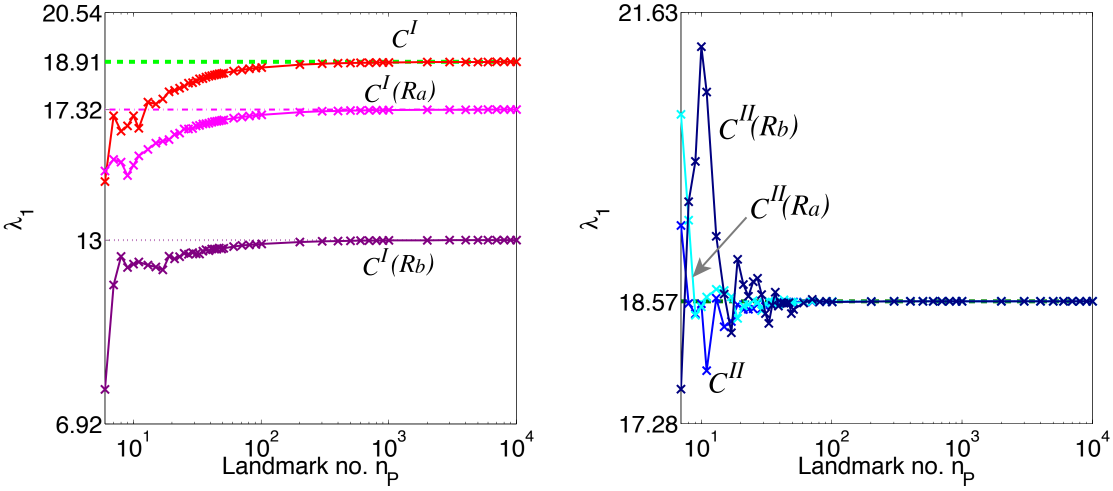
Figure 3.9. Different reparameterizations of line-bump shapes.



(a) Formulation I

(b) Formulation II

Figure 3.10. 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 Figure 3.8.



(a) Formulation I

(b) Formulation II

Figure 3.11. 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 Figure 3.8.

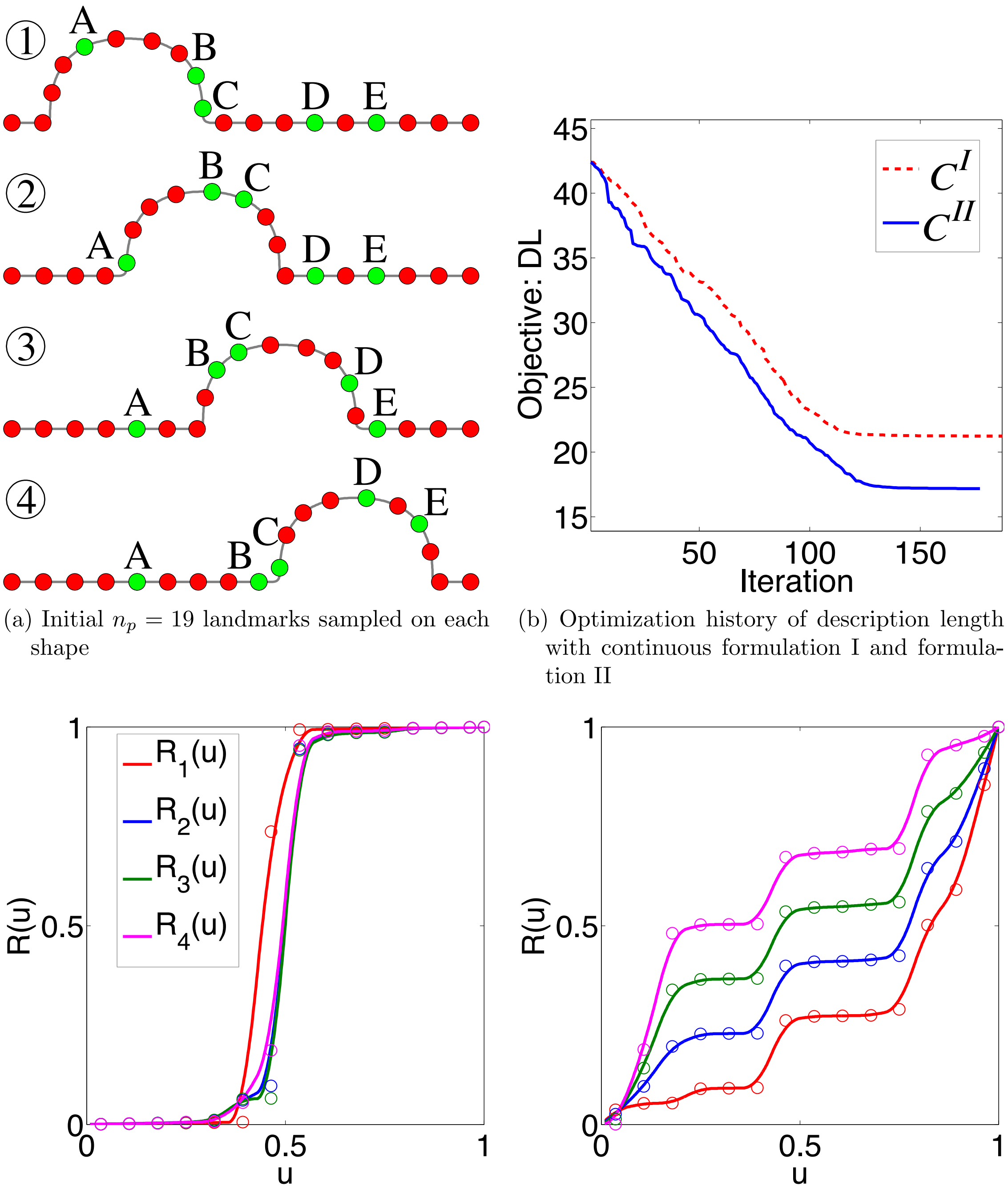
rameterizations for both formulations. The dotted straight lines in Figure 3.10(a) and Figure 3.11(a) indicating the convergence limit for continuous formulation I are computed by the analytical formula (3.30); the dotted straight lines in Figure 3.10(b) and Figure 3.11(b) are computed by using $n_G = 20$ Gauss quadrature points per knot span. Figure 3.10(a) shows, 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 shown in Figure 3.10(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 shown in Figure 3.11. This shows that continuous formulation II is parameterization independent and characterizes the intrinsic geometric property of shapes.

3.4.4 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 formulations' approximation 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 show two examples below, which illustrate the continuous formulation II's advantage over formulation I in terms of correspondence quality and faithful representation of the shapes.

3.4.4.1 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 shown in Figure 3.12(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 where bump corners do not correspond across the training set.

Correspondence optimization is conducted with the covariance matrix computed by both continuous formulation I and II's approximations. The optimization history of DL objective function is shown overlapped in Figure 3.12(b), and the optimized reparameterization functions $\{R_i(u)\}$ for formulation I and II are shown in Figure 3.12(c) and (d) respectively. It is observed that formulation I causes an almost flat platform near the two ends while formulation II does not; this suggests the 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 shown in Figure 3.13(a), where there are barely any landmarks used for representing the bump and the optimized correspondence hardly has any improvements either as revealed by the locations of the five feature land-



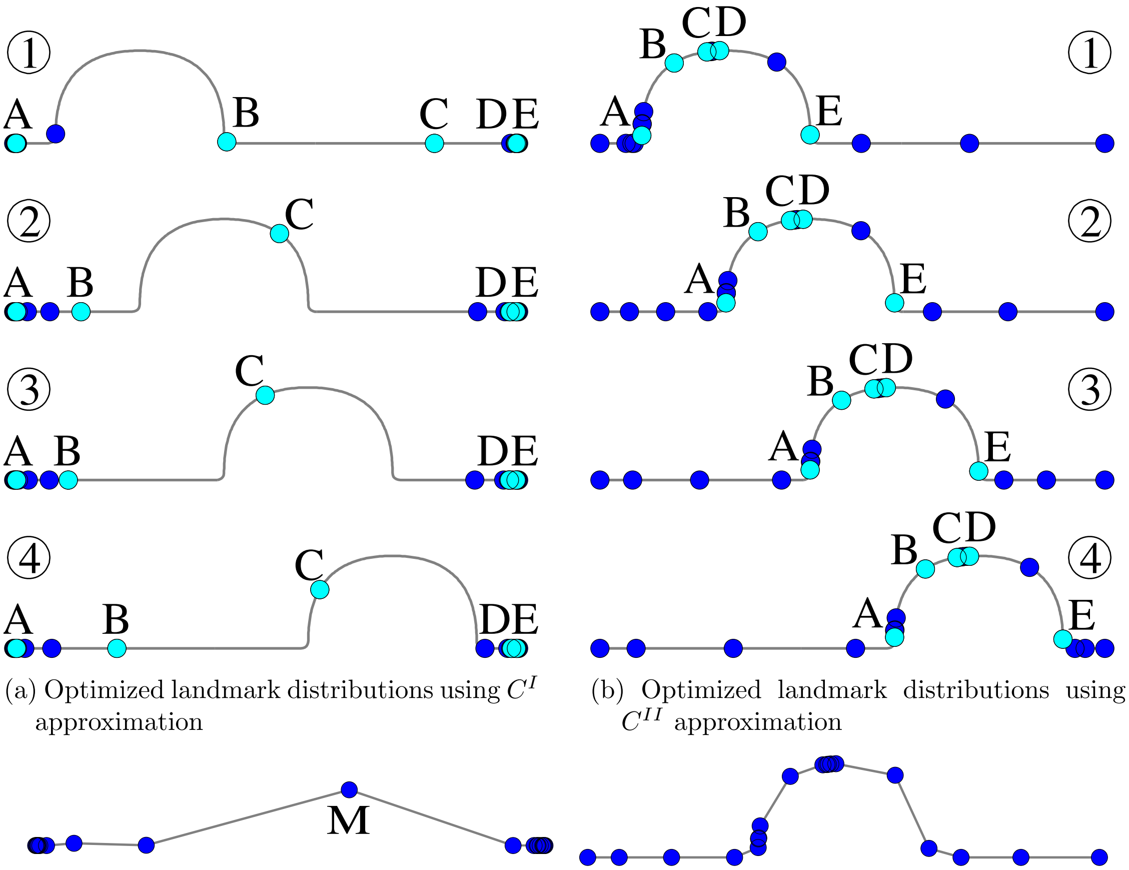
(a) Initial $n_p = 19$ landmarks sampled on each shape

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

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

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

Figure 3.12. 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}

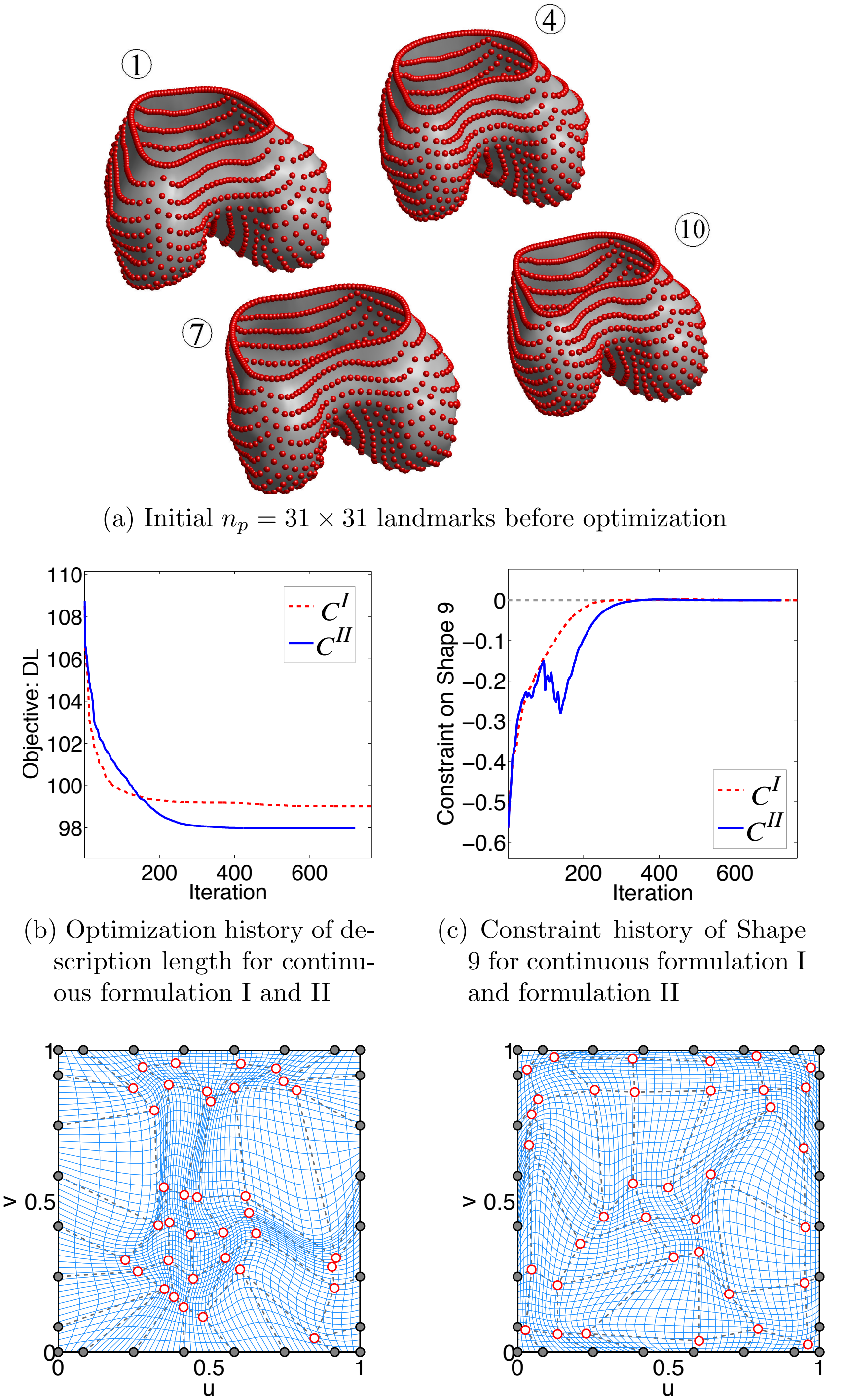
Figure 3.13. The optimized correspondence from continuous formulation I (C^I) and formulation II (C^{II}).

marks. On the other hand, optimized landmark result in Figure 3.13(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 3.13(c) and (d) respectively for formulation I and II. Recall that as a result of the difference between (1.23) and (1.25) regarding the Jacobian term $|\mathbf{J}(\mathbf{u})|$, the associated approximate formulations (3.12) and (3.14) 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 shown in Figure 3.13(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.

3.4.4.2 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. $n_P = 31\times31$ landmarks are used to represent each shape instance and $n_b = 8\times8$ control coefficients are used to model reparameterization B-spline for each shape instance.

The initial landmark configuration is shown in Figure 3.14(a). The optimization history of DL objective function for both formulations is overlapped in Figure 3.14(b). The constraint history of Shape 9, $g_9(u)$ for enforcing the positivity of Jacobian $\mathcal{J}(\mathbf{u})$ in (3.20), is shown in Figure 3.14(b), where optimized constraints become active. The optimized reparameterization function $\{R_i(u)\}$ parametric grids



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

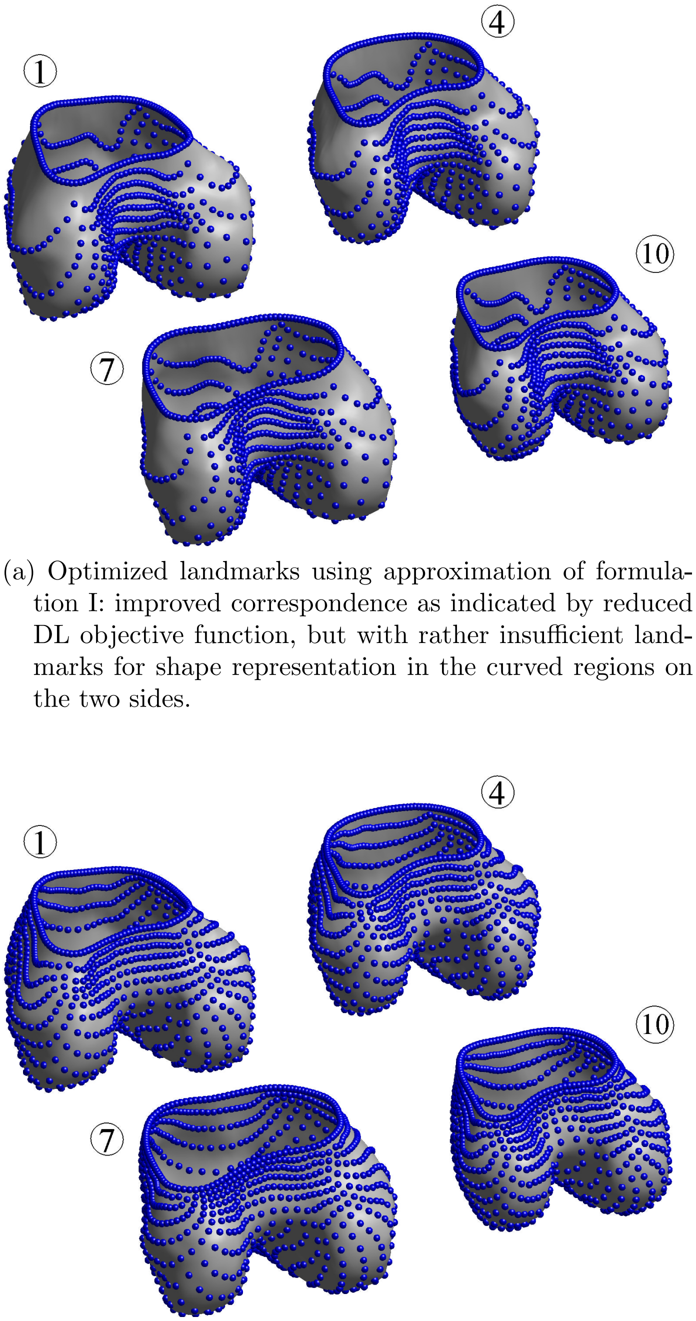
(b) Optimization history of description length for continuous formulation I and II

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

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

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

Figure 3.14. 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 3.15. 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.

for formulation I and II are shown in Figure 3.14(d) and (e) respectively.

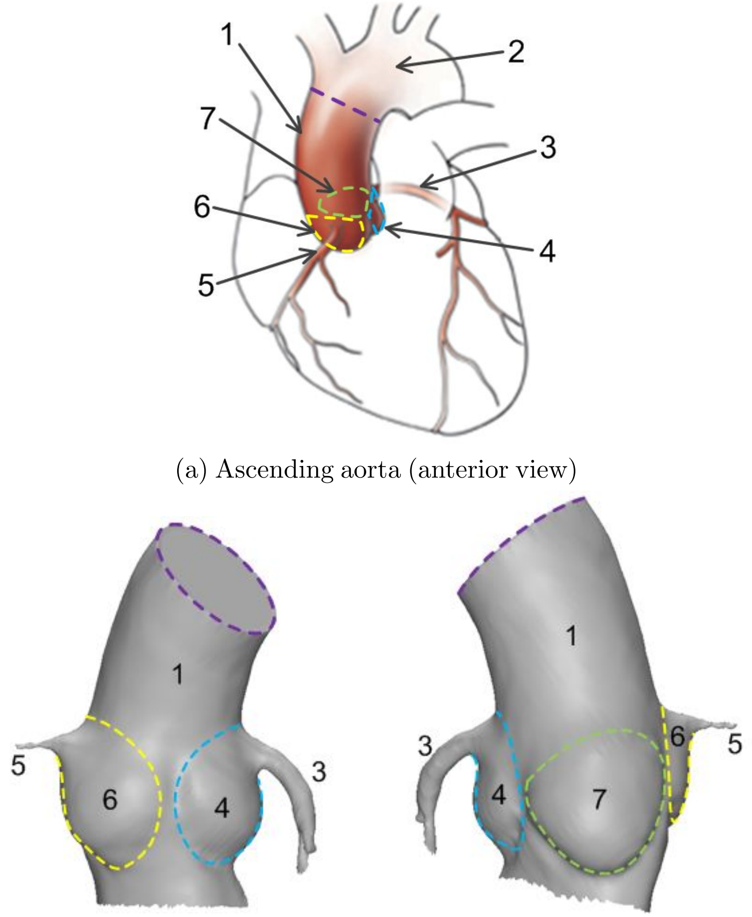
The optimized landmarks for both formulations give comparable correspondence improvements in terms of the three measures for statistical shape models (generalization ability, specificity and compactness [28]). However for formulation I, the optimized landmarks are not distributed in a way to sufficiently represent the underlying shape due to under-sampling in the bump region as shown in Figure 3.15(a). In contrast, formulation II not only gives comparable correspondence improvement, but also provides a sufficient shape sampling and reasonable optimized landmarks distribution as shown in Figure 3.15(b). This desirable feature of formulation II comes 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.

CHAPTER 4

APPLICATION TO AORTA

4.1 Introduction

4.1.1 Aortic disease study. Aortic diseases, including aneurysm of the ascending aorta and sinuses as well as calcification of the aortic valve, are significant causes of morbidity and mortality. The *ascending aorta* is the tube portion of the *aortic artery* starting from the upper base of left ventricle to the *aortic arch* where three branches



(a) Ascending aorta (anterior view)

(b) Front view

(c) Back view

Figure 4.1. Ascending aorta on heart with scanned data. Anatomical structure: 1) Ascending aorta (tube portion); 2)Aortic arch; 3) Left coronary artery; 4) Left coronary sinus; 5) Right coronary artery; 6) Right coronary sinus; 7) Non-coronary sinus.

originate from. Figure 4.1(a) (courtesy of Cleveland Clinic Foundation) gives the anatomical overview of the location of the ascending aorta along with sinuses on heart in the anterior view of human body. The tube chunk shown consists of ascending aorta 1 (number legend) and aortic arch 2, separating roughly at the purple dotted section lines. At the lower part of ascending aorta, there are three sinuses and two outgoing arteries, i.e. *left coronary artery* 3 based on top of *left coronary sinus* 4; *right coronary artery* 5 based on top of *right coronary sinus* 6; non-coronary sinus 7 (without no artery coming out). Figure 4.1(b)(c) are the front and back views of the isolated ascending aorta model in triangle mesh, where the sinuses are arteries highlighted and numbered in consistency with Figure 4.1(a). The cut at the top side of the mesh corresponds roughly to the section line between ascending aorta and aortic arch, as denoted by the purple dotted line. The front view is in general different than the anterior view, and it is chosen so that the left and right coronary 4 and 6 sinuses face straight at us at the same time. Non-coronary sinus 7 is located at the opposite side, as seen in the back view.

Aortic aneurysm can lead to dissection and rupture, two likely fatal events. Aneurysms deemed at risk are surgically repaired by removing the dilated portion of the vessel and removing it with a prosthetic graft. Aortic valve disease is also often treated with prostheses. Traditionally, the diseased native aortic valve was surgically removed and replaced with a prosthetic valve. More recently, aortic valve disease has also been treated via transcatheter aortic valve (TAV) replacement, where a prosthetic valve is deployed over the native valve leaflets. Biomechanics largely dictates the success of these various treatments. For instance, a TAV device which exerts excessive radial force may rupture the aortic root, whereas insufficient radial force may lead to device migration.

An understanding of the aortic biomechanics can offer scientific rationale to design better treatments for these conditions. Computational analyses, e.g. structural finite element (FE) and computational fluid dynamic (CFD) simulations, are particularly useful for improving prosthetic device design because numeric analysis allows for a fast and inexpensive way of analyzing conceptual designs and design optimization [61]. However, the accuracy of these simulations are highly dependent on the material properties, geometries, and boundary conditions prescribed; for the human ascending aorta and aortic root, these parameters are not easily defined and they can vary greatly from patient to patient. In the past, many groups have used idealized aortic geometries to simplify their analyses, but this will detract from the simulation accuracy.

In the past few years, clinical diagnostic imaging modalities have advanced significantly. Today, multi-slice CT, MRI, and 3D echocardiography can offer high resolution images of vasculatures that were previously unavailable. It is now feasible to utilize such medical images to accurately reconstruct 3D geometries of arteries and build computational models to perform structural analysis of the ascending aorta and aortic root wall on a patient-specific level. Such analysis can be used for pre-operative planning to determine the proper prosthetic device, size and positioning for a particular patient. Currently, these type of decisions are made primarily based on the physician's intuition and experience.

The caveat is that the generation of patient-specific computational models can be time-consuming. Often the anatomical geometries extracted from the clinical imaging data are not suitable for computational analysis, i.e. the surface data is too noisy or contains artifacts or holes, due to poor image resolution. As a result, significant post-processing of the 3D geometry data is often necessary. In order for the simulation-based pre-operative planning to be realistic in the clinical setting, this process must be expedited.

4.1.2 Rationale for statistical modeling of aorta. One possible solution to this problem is to create statistical shape models (SSMs) for the ascending aorta and aortic root. The mean ascending aorta and aortic root geometry could be mapped directly to the raw surface data obtained from the clinical images for each new patient, which will greatly reduce user input and consequently the time to complete the 3D geometry reconstruction process for the incoming patient. Aortic SSMs will also facilitate future probabilistic studies of the aortic biomechanics.

While patient-specific analyses are essential for accurate pre-operative planning, population-based probabilistic studies will be pivotal in the design of reliable valve and vessel prostheses and implantation techniques. The design of these devices should be robust to account for uncertainty in the tissue properties and anatomical geometries to avoid clinical adverse events and clarify patient selection criteria. Probabilistic computational analysis permits a rigorous quantification of various uncertainties and has been successfully applied to the design and analysis of a variety of engineering systems, including space vehicles and automobiles [62], and more recently, orthopedic implants [63, 64, 65, 66]. In a probabilistic ascending aorta and aortic root computational model, the anatomical geometry will be defined as a random variable with shape variation defined from the aortic SSMs.

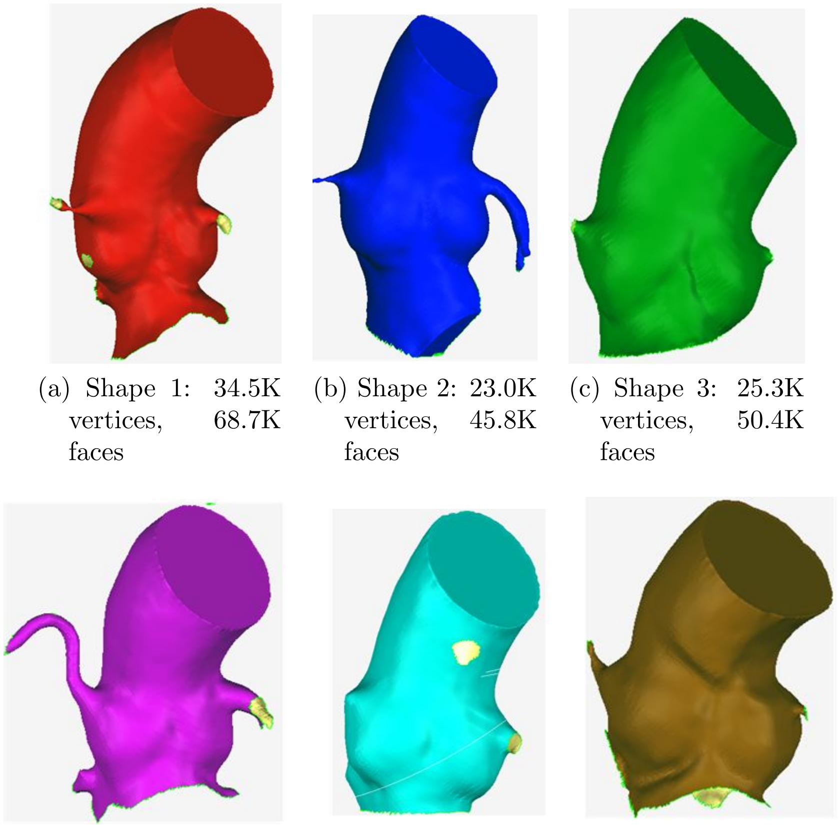
4.1.3 Related work. Statistical shape modeling is a powerful tool to capture the shape variation pattern across a group of shapes belonging to a certain shape class [1]. SSM has seen many promising applications in a great variety of medical fields such as image analysis [67], image segmentation [68][2], organ/bone shape reconstruction [69][70], treatment tracking [71], patient-specific simulation [72][73], diagnostics [74][75][76] and femoral allograft [77] and cam femoroacetabular impingement [78].

The utility of the statistical model relies on a sufficiently large training set data pool, and more importantly, a reasonably good correspondence across the entire training set. A good quality of the SSMs is usually obtained via manipulating correspondence across the shape populations in order to optimize some quality metric, e.g. the description length of the resulting SSM [19] [79]. The optimization is currently done via concatenation of multiple re-parametrization functions as proposed in [28], which is subject to undesirable computational efficiency. We propose a direct reparametrization scheme with better efficiency and use it in the statistical modeling for aorta in this thesis. Our reparametrization function is represented by B-spline coefficients with diffeomorphic constraints to guarantee a valid deformation field in the parameter domain. The computation efficiency is further improved with a differentiable representation of training set shapes and the use of adjoint method for computing analytical gradients with respect to optimization variables.

4.2 Training set data preprocessing

4.2.1 Mesh preprocessing. The raw input ascending aorta of six shape instances are shown in Figure 4.2 with distinct colors. They all contain the ascending aorta tube portion as the major part, but vary in the detailed features for sinuses and particularly for coronary arteries. Since the outgoing arteries contain incomplete geometric data due to unfavorable scan conditions (Shapes 1,3,4,5,6 missing left coronary artery, Shape 1,2,3,5,6 missing right coronary artery), they will be excluded from major tube part and we will focus on the sinus part and the tube portion. Table 4.1 lists the IDs of the six contributing patients along with their respective gender and age information. In order for the data to be processable by our proposed algorithm, the raw training set must go through a series of preprocessing procedures that resolve issues such as incomplete data, noise and smoothness etc.

4.2.1.1 Hole filling. Incomplete data is a common issue from reconstructed mesh from CT scans. This usually comes in the form of holes, which can be grouped into two types depending on the requirements of hole filling: 1) direct filling; 2)



(a) Shape 1: 34.5K vertices, 68.7K faces

(b) Shape 2: 23.0K vertices, 45.8K faces

(c) Shape 3: 25.3K vertices, 50.4K faces

(d) Shape 4: 25.1K vertices, 49.7K faces

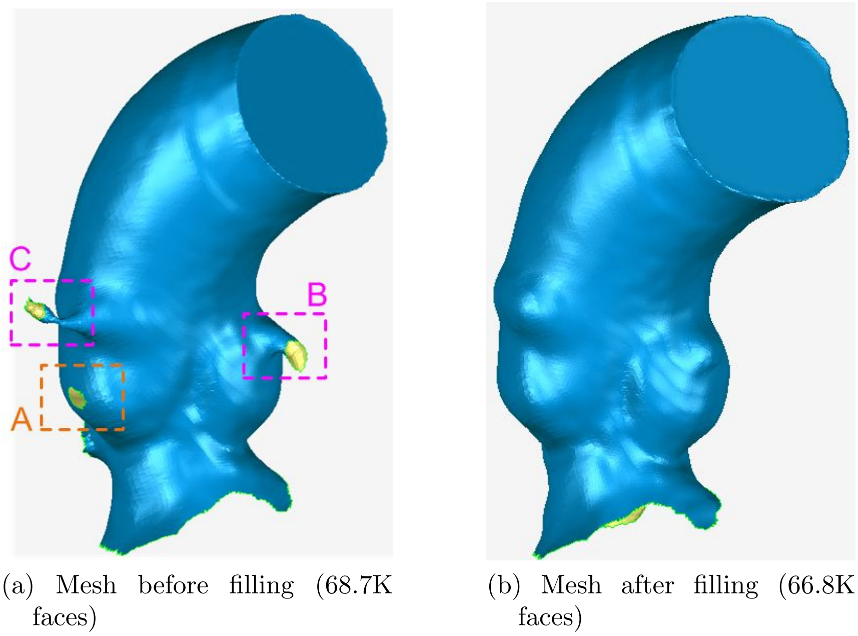
(e) Shape 5: 21.1K vertices, 41.9K faces

(f) Shape 6: 21.3K vertices, 42.0K faces

Figure 4.2. Raw triangle meshes of aorta data in front view

Table 4.1. Metadata of raw aorta models

Shape	Patient ID	Gender	Age
1	1955746	Male	44
2	1986856	Male	42
3	1996415	Male	59
4	2066014	Male	37
5	2073023	Male	45
6	2214341	Male	49

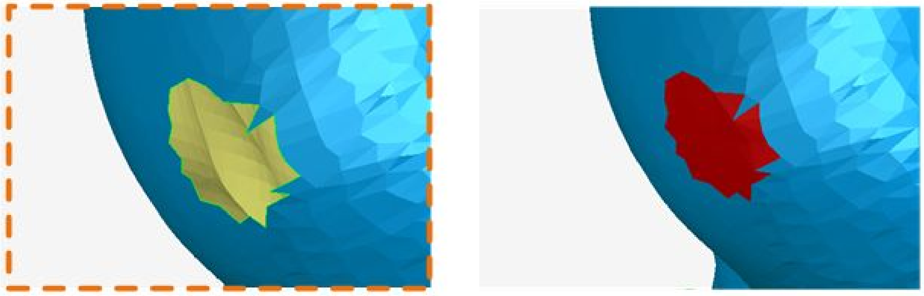


(a) Mesh before filling (68.7K faces)

(b) Mesh after filling (66.8K faces)

Figure 4.3. Mesh hole filling and hole types. Type 1: direct filling (hole A); type 2: flattening and filling (hole B and C)

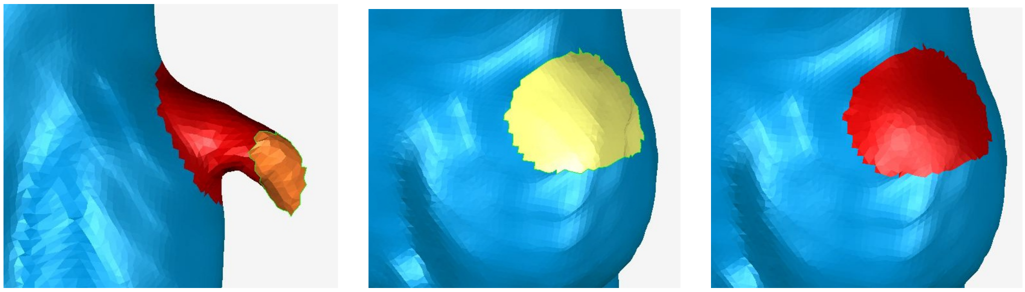
hole flattening and filling. Type 1 is usually minor data loss and has mild curvature variation in the vicinity; Type 2 is usually associated with a cut-off artery stemming from the sinus; in this case, direct filling of hole cannot recover the geometry at the cut-off location around the thin artery tube. The little influence of the artery tube also makes it reasonable to remove the artery feature and seal the flattened hole off before doing any direct hole filling.



(a) Hole A: before filling

(b) After filling

Figure 4.4. Direct hole filling for type 1 (hole A)



(a) The dangling feature to remove for hole flattening

(b) Flattened holes

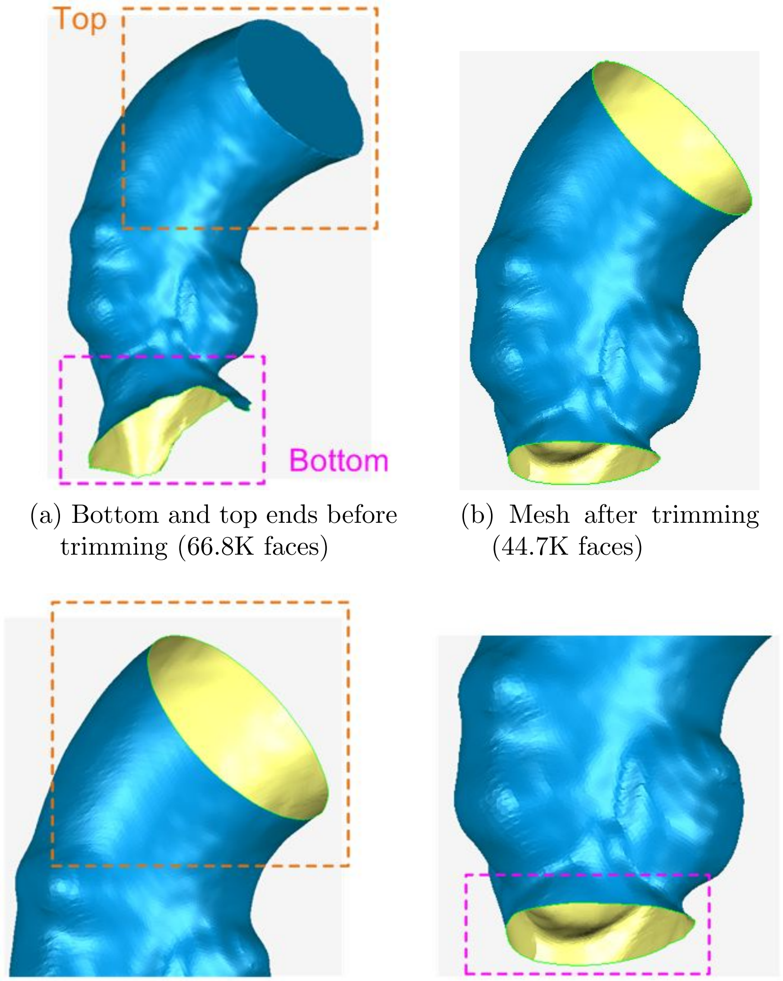
(c) Holes filled

Figure 4.5. Flattening and filling for type 2 (hole B)

Taking Shape 1 as an example in Figure 4.3(a), hole A located on the right coronary sinus belongs to Type 1, which can be directly filled by mesh completion algorithm as shown in Figure 4.4. The other two holes B (left coronary artery) and C (right coronary artery) are of Type 2, where a feature removing step is necessary before mesh completion algorithm is applied. Figure 4.5(a) shows the feature portion around hole B expected to be eliminated, and a flattened hole in Figure 4.5(b) follows, and in the end the mesh completion based on curvature constraint will result in the filled hole show in Figure 4.5(c). Similarly hole C could be flattened and filled in the same way, and the aorta mesh after the hole filling procedure is shown in Figure 4.3(b).

4.2.1.2 End trimming. The proposed algorithm currently can only handle training set geometry topologically equivalent to a disc. The geometry in Figure 4.3(b) after the hole filling is complete satisfies such criterion but wasted a rather large area at the top end since the planar cut between ascending portion and aortic arch carries little variation of the aortic tube surface. Therefore it is advisable to trim the top end to an open end and map the remaining cylindrical part onto a square domain. The top end trimming occurs as shown in Figure 4.6(c) compared to the original state in Figure 4.6(a).

For the bottom part, the factor of noise and data separation all make the bottom portion unreliable to faithfully represent the lower part of the ascending aorta. The irregular mesh boundaries observed for Shape 1, 2, 4 and 6 in Figure 4.2(a)(b)(d)(f) have confirmed this need for the bottom end trimming. Additionally, when the bottom portion lacks data to be trimmed, as is the case for Shape 3 in Figure 4.2(c), the mesh boundary must first be extended to create enough data to be trimmed based on the bordering mesh's curvature information. The mesh after bottom end trimming will look like Figure 4.6(d) where the mesh boundary becomes much more regular and smooth. After the top and bottom trimming, the aorta mesh



(a) Bottom and top ends before trimming (66.8K faces)

(b) Mesh after trimming (44.7K faces)

(c) Trimming at top end

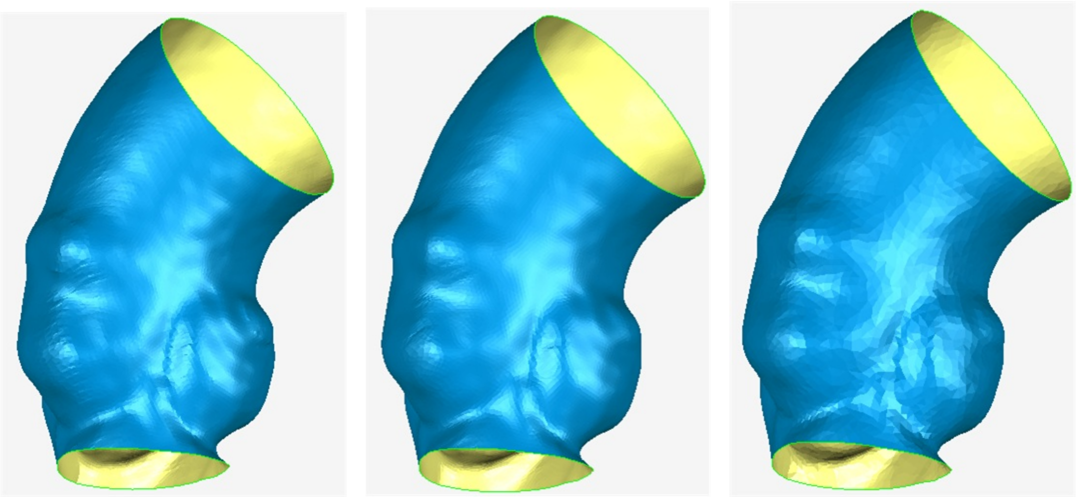
(d) Trimming at bottom end

Figure 4.6. Trim mesh at top and bottom

becomes Figure 4.6(b) and the number of faces drops from 66.8K to 44.7K.

Since the correspondence of the boundaries are assumed to be fixed and only the interior correspondence is optimized in our approach, it is also desirable that the manual trimming position of all the six shapes are at approximately the corresponding location across the entire training set.

4.2.1.3 Smoothing and decimation. The illustrative data obtained till trimming is shown in Figure 4.7(a). To further reduce noise, we apply a smoothing filter on the mesh based on local mesh curvature to obtain the smoothed mesh in Figure 4.7(b), where the number of faces drops from 44.7K to 37.4K. For a more compact representation, the smoothed mesh is further decimated from 37.3K to 10.1K.



(a) Mesh after trimmed (22.7K vertices, 44.7K faces) (b) Mesh after smoothing (18.9K vertices, 37.4K faces) (c) Mesh after decimation (5.1K vertices, 10.1K faces)

Figure 4.7. Mesh smoothing and decimation

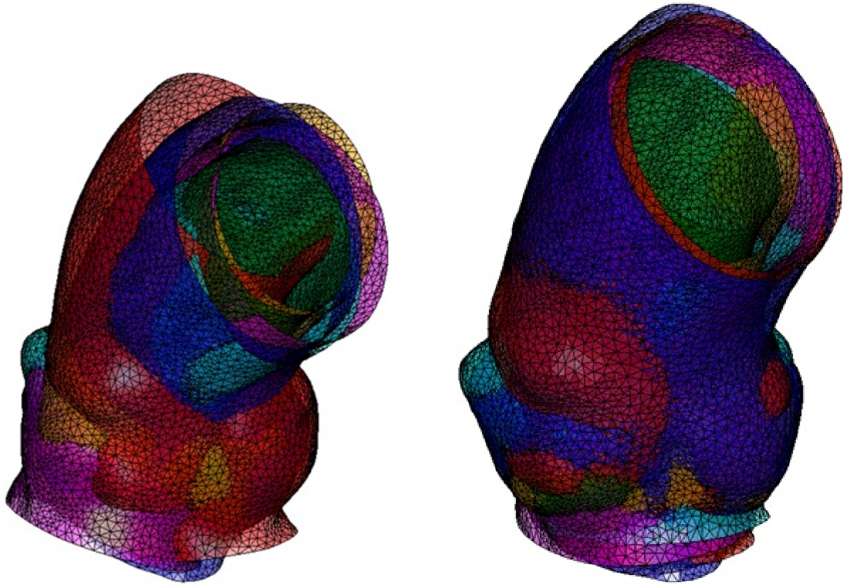
During the entire mesh pre-processing procedures comprising hole filling, end trimming, smoothing and decimation, the number of vertices and triangle faces in general will change. Table 4.2 keeps track of this information as each procedure progresses.

Table 4.2. Vertex and triangle numbers during SSM pre-processing.

Vertex# (Face#)	Raw	Filled	Trimmed	Smoothed	Decimated
Shape 1	34569 (68717)	33531 (66838)	22774 (44734)	18927 (37405)	5116 (10099)
Shape 2	23049 (45828)	20739 (41300)	16814 (32871)	13994 (27571)	5162 (10199)
Shape 3	25345 (50441)	25939 (51766)	20945 (41002)	17324 (34146)	5185 (10243)
Shape 4	25102 (49714)	22861 (45558)	18823 (36745)	15547 (30591)	5117 (10094)
Shape 5	21097 (41940)	21431 (42764)	17882 (30540)	14402 (28409)	5179 (10227)
Shape 6	21271 (42047)	20903 (41663)	17242 (33651)	14436 (28469)	5188 (10248)

4.2.2 Training set B-spline fitting.

4.2.2.1 One-time mesh alignment. In the correspondence optimization, the alignment process is discarded for additional speed gain. This requires the shapes

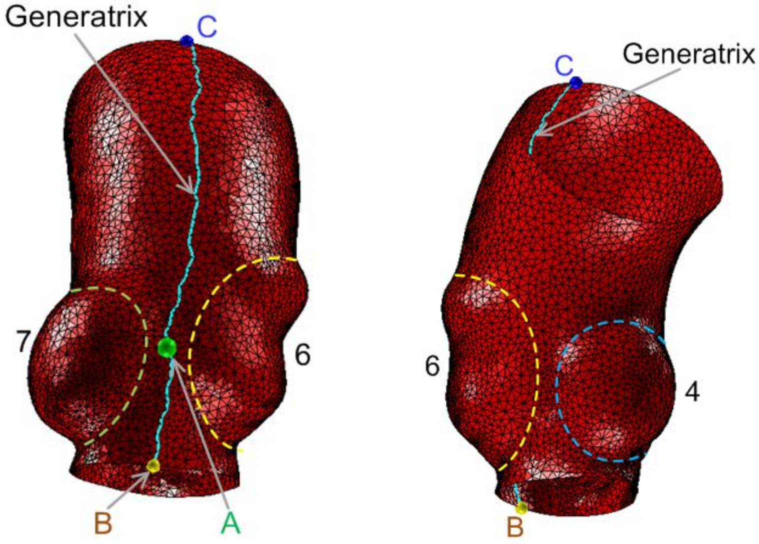


(a) Before alignment

(b) After alignment

Figure 4.8. six meshes before and after one-time alignment with ICP algorithm should already be aligned before they are fed into the optimization algorithm. Therefore, we perform a one-time alignment of the triangle meshes before the actual optimization. As suggested by the last column in Table 4.2, the number of vertices of all aorta meshes after the mesh preprocessing is not identical across the trainings set, and the Iterative Closest Point (ICP) algorithm is employed to align these meshes. Before alignment, the six triangle meshes are superimposed and shown in Figure 4.8(a). The ICP aligned meshes of training set are shown in Figure 4.8(b), where the six meshes are brought to a much better alignment state.

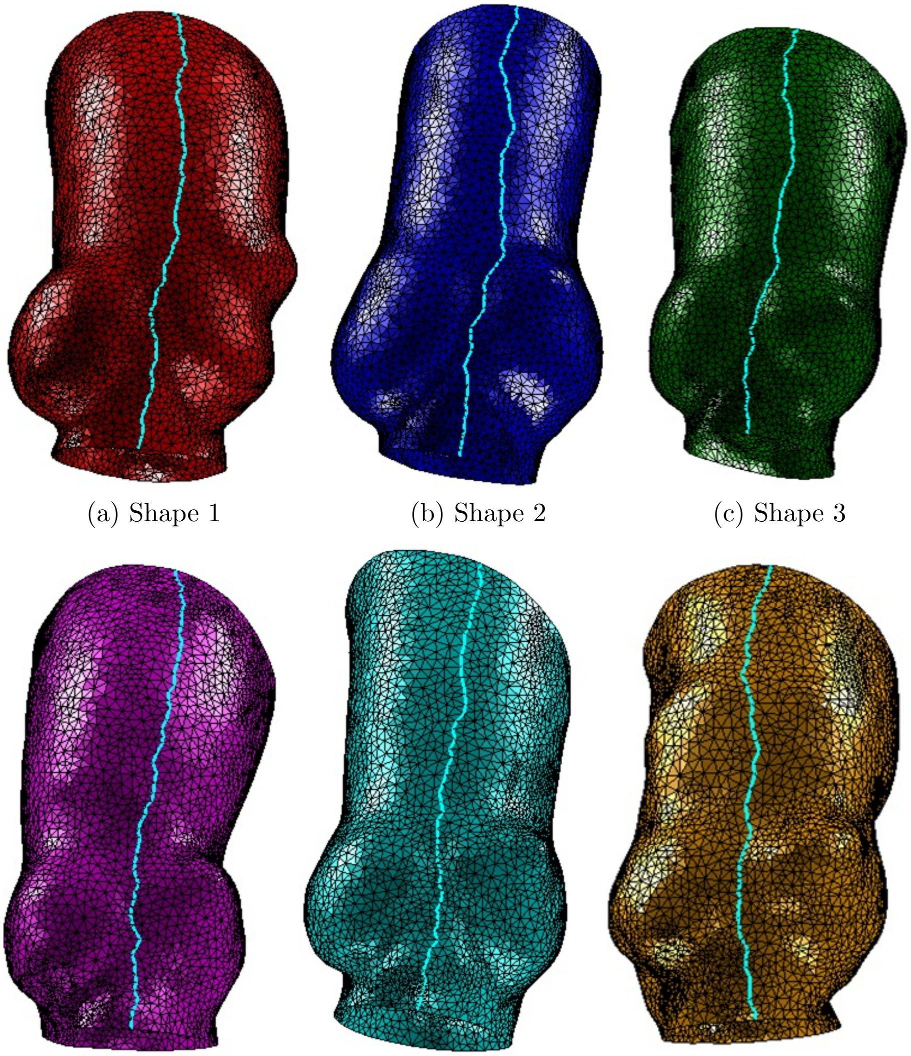
4.2.2.2 Generatrix determination. Recall that our SSM algorithm requires a square domain, whereas the current triangle mesh is topologically equivalent to a cylinder. One simple way to resolve this topological discrepancy is to cut the mesh along a line and unfold and map it onto the square domain; this cutting line is the "generatrix" to be determined. Since it is better to leave the area between the left and right coronary sinuses (i.e. 4 and 6) intact from whatever influence brought by the cutting line, the cutting line is placed on the back side between the right coronary



(a) Left view (b) Front view

Figure 4.9. Generatrix determination. Number legend: 4). Left coronary sinus; 6). Right coronary sinus; 7) Non-coronary sinus

sinus 6 and the non-coronary sinus 7 as shown in Figure 4.9(a).



(a) Shape 1 (b) Shape 2 (c) Shape 3

(d) Shape 4 (e) Shape 5 (f) Shape 6

Figure 4.10. Generatrix on all the six shapes in left view

Again taking Shape 1 as an example, the first step is to manually specify vertex A, which lies at a valley point between 6 and 7. Then the Dijkstra's algorithm is utilized to find a vertex at the bottom and top boundaries with the shortest geodesic distance to A along mesh edges. Comparing within boundary vertices at bottom and top side gives vertex B and C; connecting A with both B and C results in line BAC as the generatrix shown in Figure 4.9(b) in the front view. Repeating these steps give the generatrix lines for the remaining five shapes as displayed in Figure 4.10.

4.2.2.3 Mesh parametrization. After the generatrix is available for each shape, it is safe to cut along it, and the generatrix edge line on the mesh will serve as the image mapped to two opposite sides of the square parameter domain. The generatrix and bottom and top boundaries are highlighted in green in Figure 4.11(a). The two vertices of the generatrix at bottom and top serve as the four corner vertices of the square domain as shown in Figure 4.11(b).

Suppose each raw shape instance $\mathbb{TD}_i$ is represented by a triangulated mesh consisting of vertex list $\mathcal{V}_p = \{\mathbf{p}_j\}(j = 1, ..., n_v)$ with associated triangle list $\mathcal{T} = \{\mathbf{T}_k\}(k = 1, ..., n_t)$, where the j-th vertex is $\mathbf{p}_j = [x_j, y_j, z_j]$. The k-th triangle is of vertex index set $\mathbf{T}_k = [\tau_{k1}, \tau_{k2}, \tau_{k3}]$, and n_v and n_t are the number of vertices and triangles respectively. The mesh parametrization procedure contains two steps: 1) initial parametrization; 2) parametrization improvement by minimizing distortion.

Initial parametrization. Mesh parametrization seeks to find a mapping of vertices between the physical domain and the parameter domain i.e. $\mathcal{V}_U = \{\mathbf{u}_j\} = \{u_j, v_j\}$. If a square parameter domain is chosen, the n_b boundary vertices parameters set $U_b = (\mathbf{u}_{n_e+1}, ..., \mathbf{u}_{n_v})$ on the four sides could be either manually determined by known correspondence, or by the correspondence manipulation for 2D case, where $n_e = n_v - n_b$ is the number of interior vertices. The standard approach to obtain a mesh parametrization without fold-over is to consider the edges of the triangle mesh to be a spring web connected at vertices [80]. In this spring model, the minimum spring energy state is reached when each interior parameter point $\mathbf{u}_j$ is an affine combination of its neighbors, i.e.

$$\begin{aligned} \mathbf{u}_j &= \sum_{k \in N_j} w_{jk} \mathbf{u}_k, \\ \text{s.t.} \quad & \sum w_{jk} = 1, \end{aligned} \tag{4.1}$$

where w_{jk} is the normalized weight coefficients of the neighbor set N_j for the j-th parameter point. Separating interior and boundary vertices gives

$$\mathbf{u}_j - \sum_{k \in N_j, k \leq n_e} w_{jk}\mathbf{u}_k = \sum_{k \in N_j, k > n_e} w_{jk}\mathbf{u}_k. \tag{4.2}$$

This reduces to solving two linear systems

$$AU = \bar{U} \quad \text{and} \quad AV = \bar{V}, \tag{4.3}$$

where U and V are the interior parameters to solve and $\bar{U}$ and $\bar{U}$ are the boundary parameter conditions; $n_e \times n_e$ weight coefficient matrix $A = (a_{ij})_{i,j=1,\ldots,n_e}$ has elements

$$a_{ij} = \begin{cases} 1 & \text{if } \; j = k \\ -w_{jk} & \text{if } k \in N_j \\ 0 & \text{otherwise} \end{cases} \tag{4.4}$$

Parametrization distortion minimization. There exist several options for assigning weight coefficients including constant (mesh geometry irrelevant) [57], and other geometry-aware barycentric coordinates such as *Wachspress*, *Discrete harmonic* and *Mean value* coordinates. These weighting options all lead to a valid parametrization but in general the mesh distortion introduced is rather high. Therefore, a mesh distortion reduction procedure should follow the initial parametrization. A simple method we employ comes from [58], where a mesh distortion measure called "stretch" is minimized.

Consider a mesh triangle $T^P = (\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3)$ in physical domain and its corresponding triangle $T^U = (\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3)$ in parameter domain. This defines a one-to-one mapping from the parameter to the physical domain, i.e. $\mathcal{S} : T^U \to T^P$. If the maximal and minimal eigenvalues of the metric tensor induced by $\mathcal{S}$ are denoted by $\Gamma(T)$ and $\gamma(T)$, the triangle stretch in the k-th triangle of the mapping or parametrization $\mathcal{S}$ could be characterized by

$$\sigma(T_k) = \sqrt{\frac{\Gamma^2(T_k) + \gamma^2(T_k)}{2}}\,. \tag{4.5}$$

Then the stretch of each vertex is defined by

$$\sigma(\mathbf{u}_j) = \sqrt{\frac{\sum A(T_k)\sigma^2(U_k)}{\sum A(T_k)}}, \tag{4.6}$$

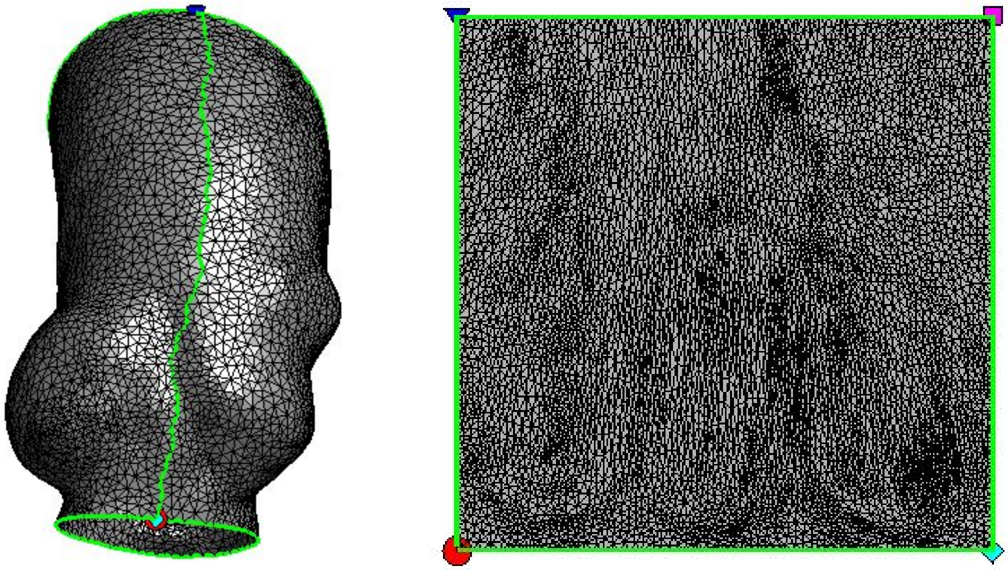
where $A(T)$ is the area of triangle T and the sums taken over all triangles T_k that have $\mathbf{p}_j$ as a vertex. The weights in (4.4) for the $(h+1)$-th iteration are updated according to the vertex stretch value at the current h-th iteration by

$$w_{jk}^{h+1} = \frac{w_{jk}^h}{\sigma(\mathbf{u}_j^h)}. \tag{4.7}$$

The stopping criterion is based on the global stretch metric defined by

$$E^h = \sqrt{\frac{\sum_{k=1}^{n_t} A_k \sigma^2(T_k^h)}{\sum_{k=1}^{n_t} A_k}}. \tag{4.8}$$

Iteration stops if $E^{h+1} > E^h$, meaning there is no room to relax the stretch and distortion. Finally $\{\mathbf{u}_j^{h+1}\}$ will be output as the optimized parameters, and the mesh parametrization $\mathcal{S}$ is established. This approach could significantly reduce the mesh distortion introduced by parametrization.



(a) Physical domain

(b) Parameter domain

Figure 4.11. Mesh parametrization

4.2.2.4 Regular sampling. With an established parametrization mapping the parameter domain to the physical domain, it is time to generate a regular grid by sampling at a regularly spaced point in the parameter domain as shown in Figure 4.12(a).

Specifically for each shape, n_M sampling parameters $\{\mathbf{m}_l\}(l = 1, ..., n_M)$ will be regularly placed within the parameter domain $[0, 1] \times [0, 1]$ where each sampling parameter point $\mathbf{m}_l = (u_l, v_l)$ results in a sampled point $\mathbf{q}_l$ on the physical mesh. The sampling is computed based on the previously obtained parametrization $\mathcal{S}$ by barycentric interpolation

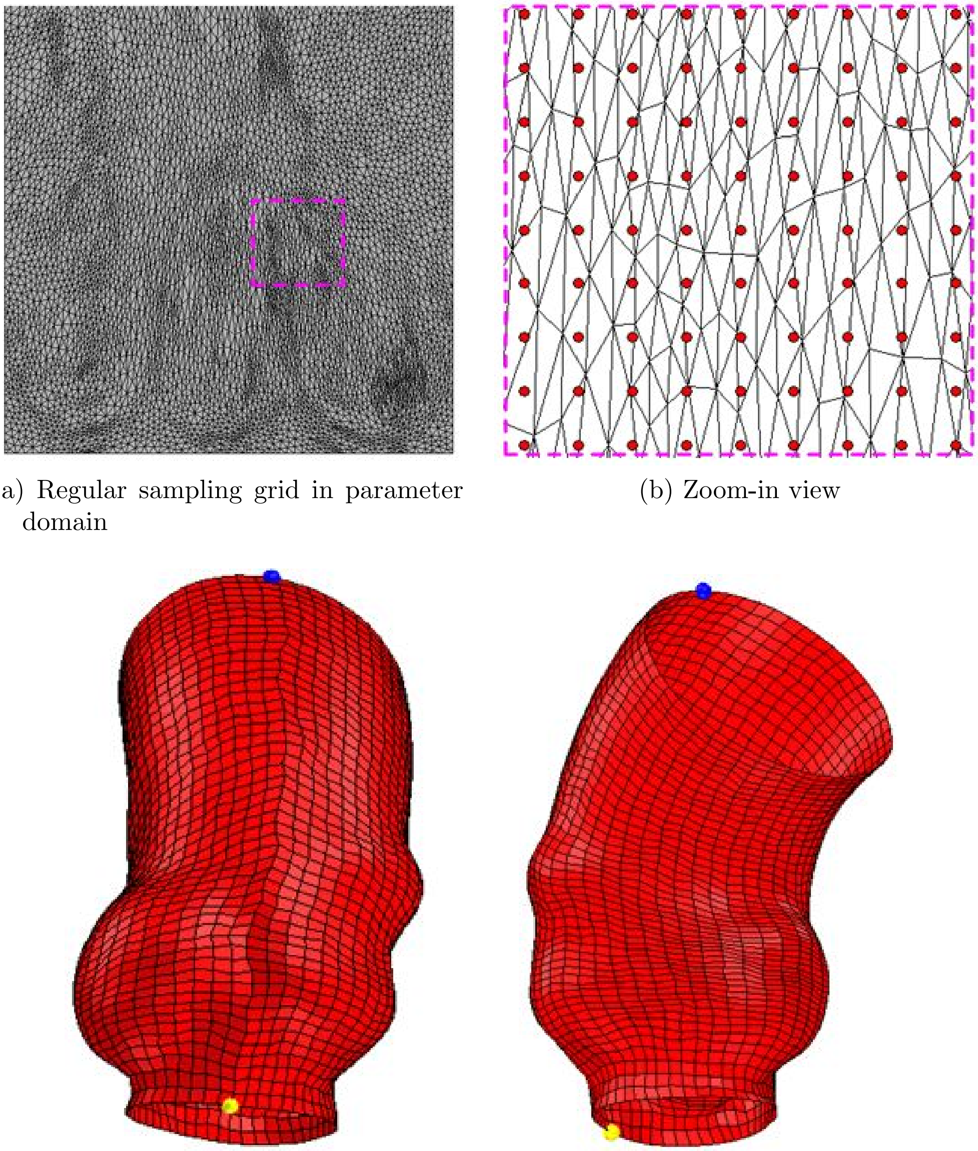
$$\mathbf{q}_l = \beta_{1,l}\mathbf{p}_1 + \beta_{2,l}\mathbf{p}_2 + \beta_{3,l}\mathbf{p}_3, \tag{4.9}$$

where the barycentric coordinates are determined by the area fractions of the three sub-triangles formed by connecting $\mathbf{m}_l$ with $\mathbf{u}_1$, $\mathbf{u}_2$ and $\mathbf{u}_3$

$$\begin{aligned} \beta_{1,l} &= \frac{A(\mathbf{m}_l, \mathbf{u}_2, \mathbf{u}_3)}{A(\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3}, \\ \beta_{2,l} &= \frac{A(\mathbf{u}_1, \mathbf{m}_l, \mathbf{u}_3)}{A(\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3}, \\ \beta_{3,l} &= \frac{A(\mathbf{u}_1, \mathbf{u}_3, \mathbf{m}_l)}{A(\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3}, \\ \text{s.t.} \quad & \beta_1 + \beta_2 + \beta_3 = 1, \end{aligned} \tag{4.10}$$

Here a regularly spaced grid of resolution 51 × 51 are sampled as seen in the zoomed-in local view Figure 4.12(b). The sampling involves interpolation of barycentric coordinates and the sampled grid in the front and left view in Figure 4.12(c) and (d). The yellow and blue spheres are the bottom and top end vertices of the generatrix line.

4.2.2.5 B-spline fitting. Base on the coordinates of the regular grid of 51×51 data points, it is time to use the global approximation technique described in [32] for fitting a B-spline surface as shown in Figure 4.13(a)(b) in the front and left view respectively. The B-spline control point resolution is set to be 30×30 and the degree along u- and



(a) Regular sampling grid in parameter domain

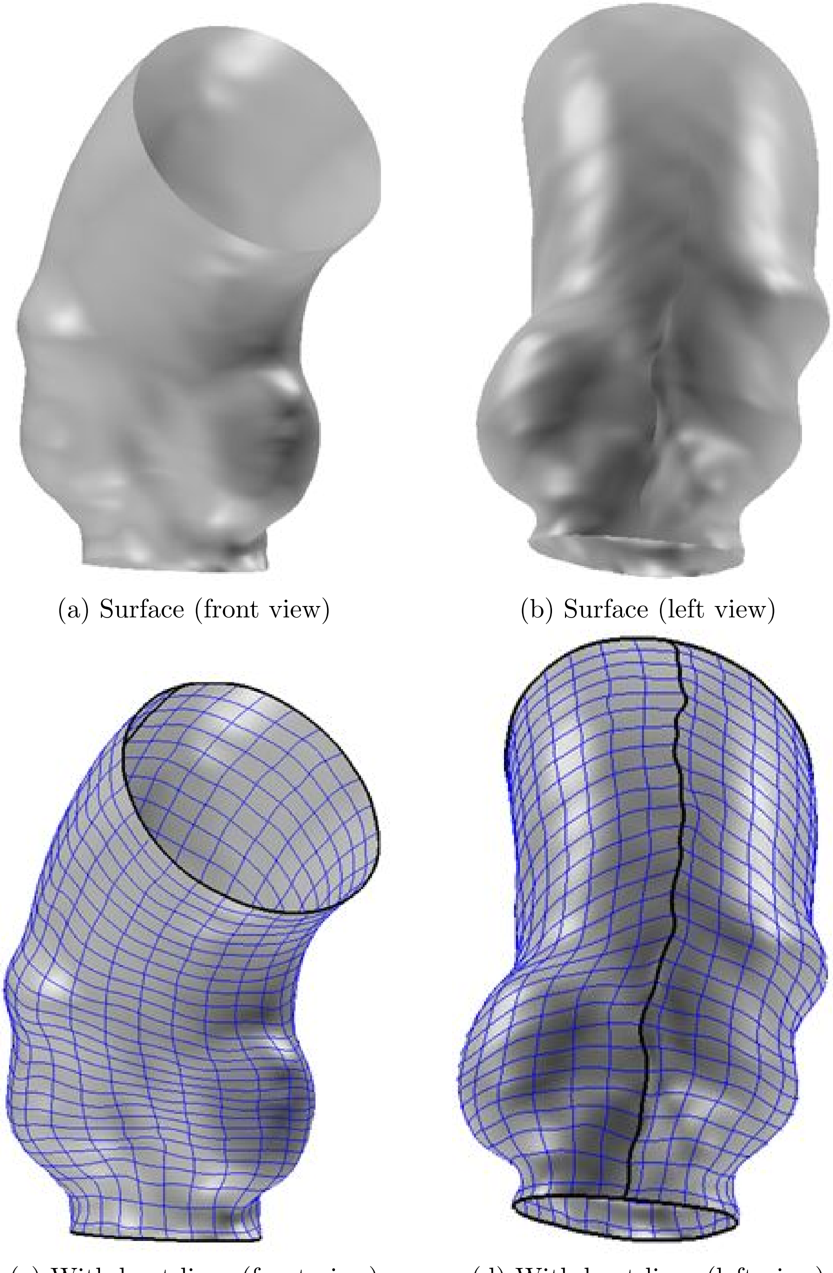
(b) Zoom-in view

(c) Sample grid in physical domain (left view)

(d) Front view

Figure 4.12. Regular grid sampling

v-direction are both quadratic. The B-spline is re-visualized with knot lines shown in Figure 4.13(c)(d). The generatrix line can be also shown in Figure 4.13(d), and the B-spline training set of six instances are shown in Figure 4.14.



(a) Surface (front view)

(b) Surface (left view)

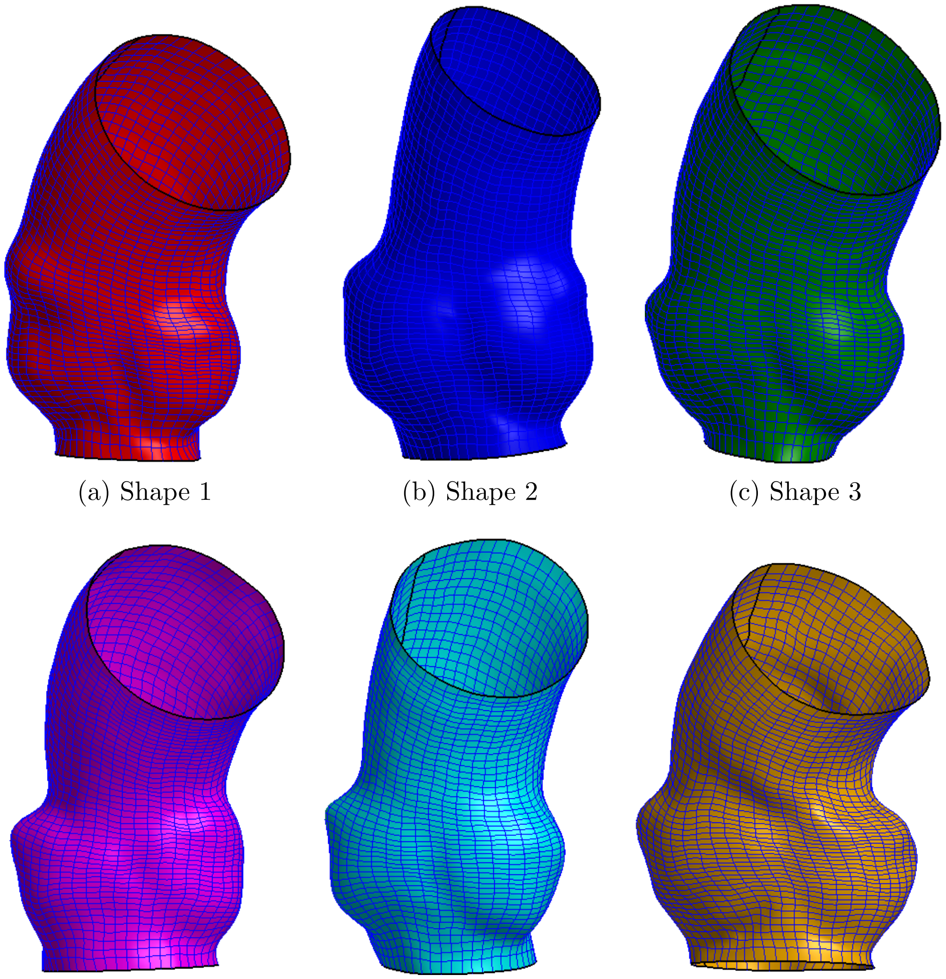
(c) With knot lines (front view)

(d) With knot lines (left view)

Figure 4.13. B-spline fitting

4.3 SSM results

The raw training set in triangle meshes have been processed into differentiable



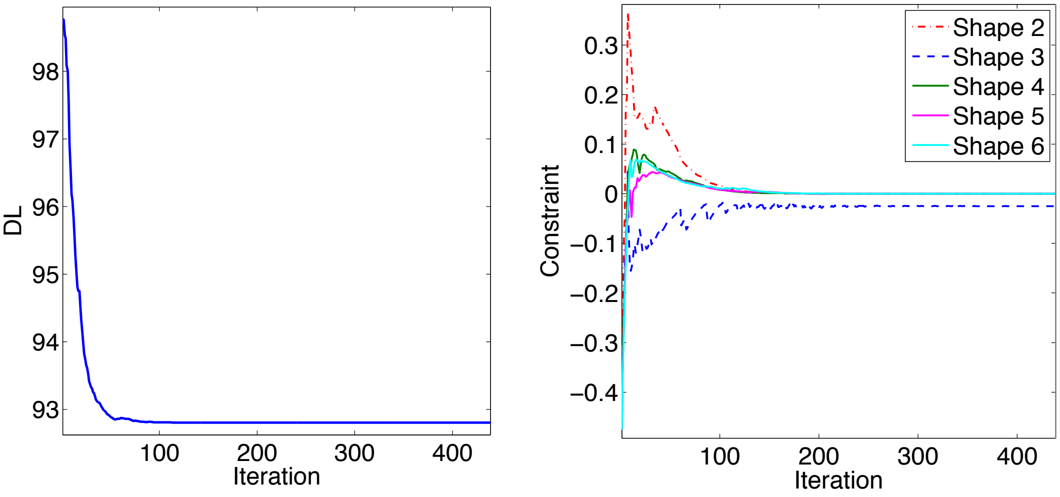
(a) Shape 1 (b) Shape 2 (c) Shape 3

(d) Shape 4 (e) Shape 5 (f) Shape 6

Figure 4.14. Training set B-splines

B-spline representation. The reparametrization function is governed by a bi-quadratic B-spline at coefficient resolution of 12 × 12. The reference shape is chosen to be Shape 1, whose sampled landmarks remain fixed throughout the optimization. The optimization is achieved by the Sequential Quadratic Programming (SQP) optimizer. The convergence criterion is that the relative change in DL objective function at the k-th iteration drops below a threshold ε, i.e. $\left|\frac{DL^{(k)}-DL^{(k-1)}}{DL^{(0)}}\right| < \varepsilon$, and here $\varepsilon = 10^{-6}$. The optimization takes 439 iterations and 2293.9 seconds to converge, and the DL drops from 98.7 to 92.8.

The history of aggregated constraint for all the non-reference Shapes 2,3,4,5,6 is shown in Figure 4.16(b). It is seen that at the last iteration, some of the constraints are active, and a previously violated constraint will be rectified to valid to ensure the diffeomorphism and prevent self-intersection of reparametrization function.

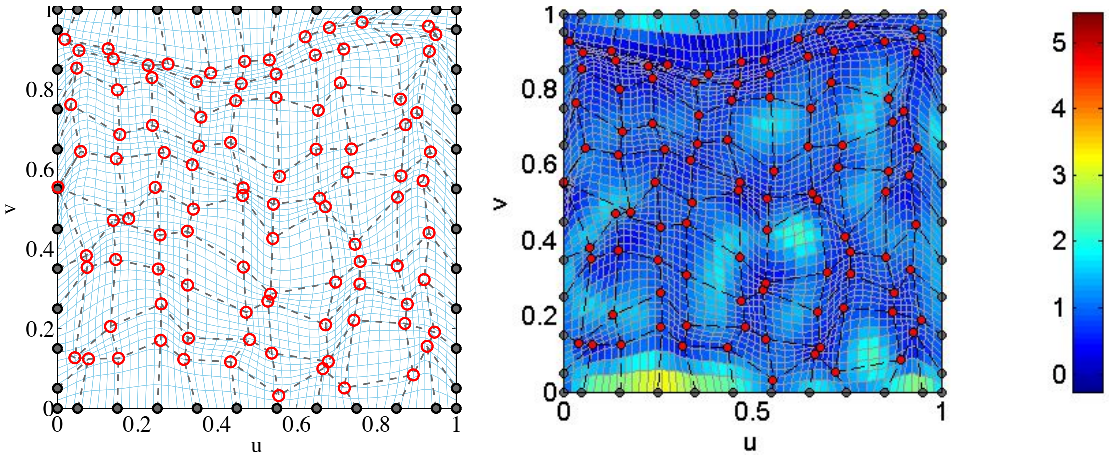


(a) Objective function DL

(b) Aggregated constraints

Figure 4.15. Optimization history

The deformed grid under the optimized reparametrization function for Shape 2 $\mathbf{D}_2(\mathbf{u})$ is shown in Figure 4.16(a), and it is seen that the interior B-spline control coefficients while the boundary B-spline coefficients are fixed. Figure 4.16(b) displays the corresponding Jacobian field.



(a) Deformed landmark grid under the optmized $\mathbf{D}_2(\mathbf{u})$

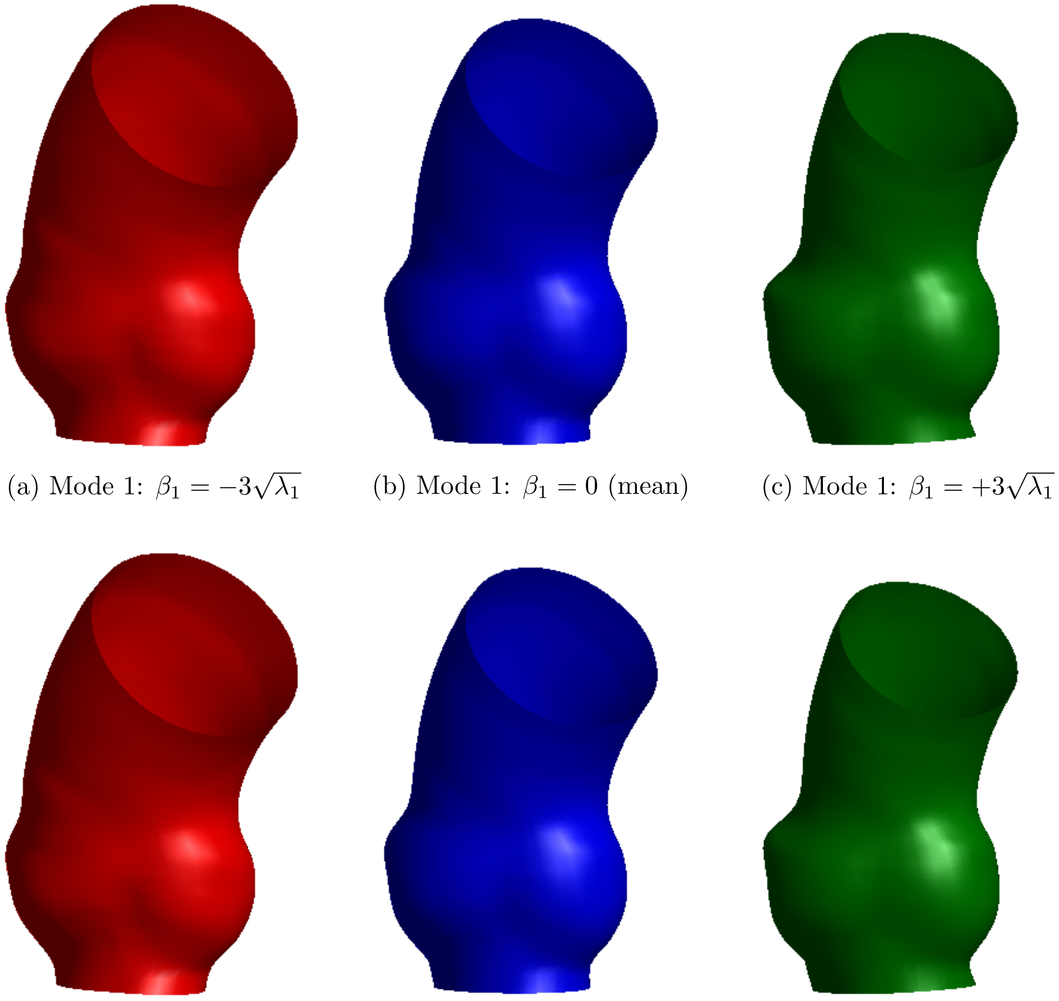
(b) Jacobian field with reparmetrized parameter domain

Figure 4.16. Deformed grid under optimized reparametrization function $\mathbf{D}_2(\mathbf{u})$

The first two modes are shown in Figure 4.17 where β is the parameter value along mode directions and λ is the variation along mode direction. The parameter values is chosen at $\beta = -3, 0, +3$ since the $[-3, +3]$ range will cover 97.7% of the possible shape variation along a particular mode. It is seen that after the correspondence optimization, the first two modes together account for 78.3% of the total variation of all possible variation patterns. Mode 1 and 2 have characterized the changing of diameter of the aorta tube and also the size changing at the left coronary sinus.

Lastly, we use the three statistical model evaluation measures [11], i.e. generalization, specificity and compactness error to analyze the effectiveness of our proposed algorithm.

The generalization error measures the ability of the SSM to extrapolate any valid instance that is not a member of the training set but belongs to the shape class. The error is calculated by the *leave-one-out* test, where each instance is left out and the remaining $n_S - 1$ shapes are used to generate statistical model and the left-out instance is reconstructed by projecting it onto the eigenmode directions, and finally the error is identified as the difference between the left-out instance and its recon-



(a) Mode 1: $\beta_1 = -3\sqrt{\lambda_1}$

(b) Mode 1: $\beta_1 = 0$ (mean)

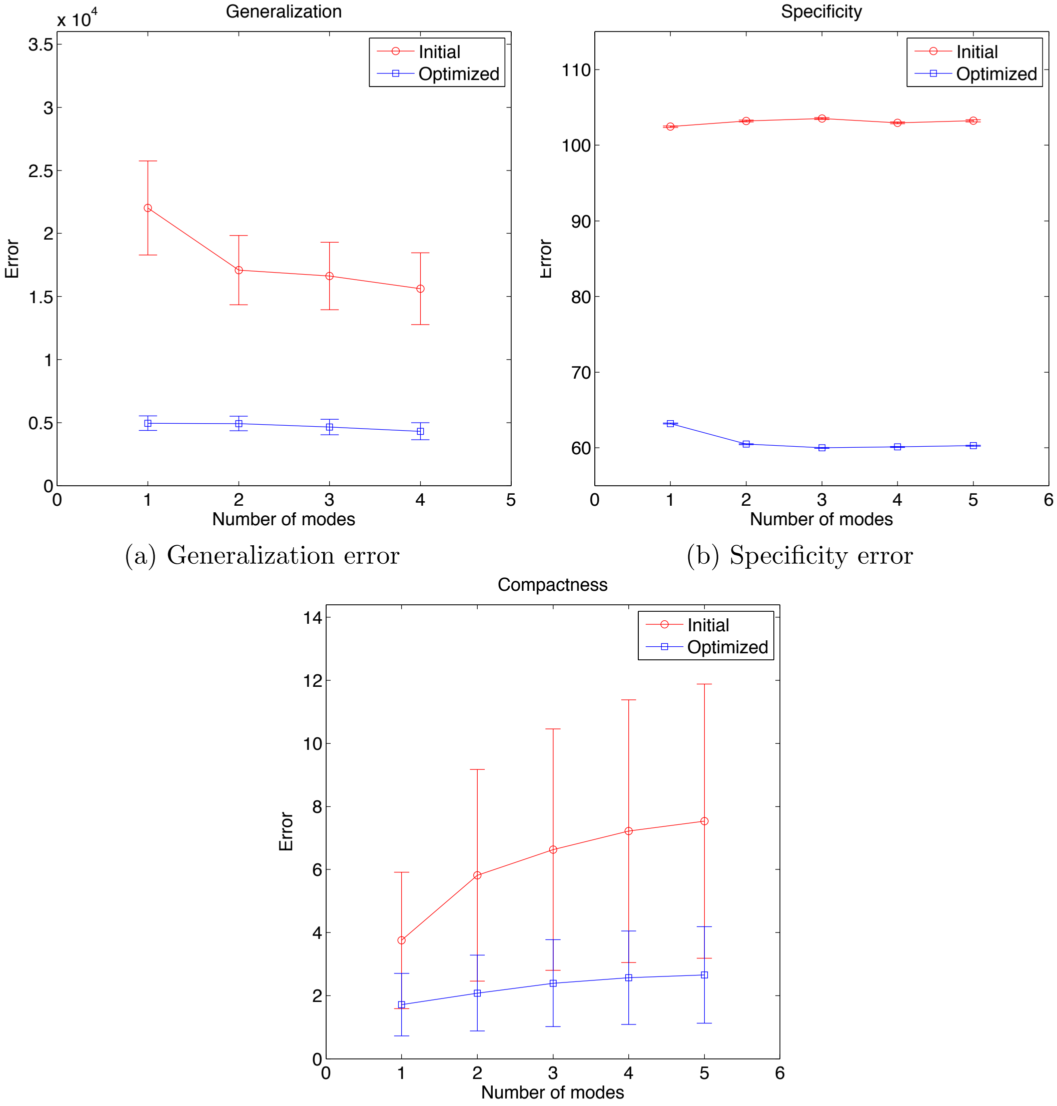
(c) Mode 1: $\beta_1 = +3\sqrt{\lambda_1}$

(d) Mode 2: $\beta_2 = -3\sqrt{\lambda_2}$

(e) Mode 2: $\beta_2 = 0$ (mean)

(f) Mode 2: $\beta_2 = +3\sqrt{\lambda_2}$

Figure 4.17. The first 2 modes of statistical model for distal femur data after optimization. $\lambda_1 = 1.71(64.5\%), \lambda_2 = 0.37(13.8\%)$.



(a) Generalization error

(b) Specificity error

(c) Compactness error

Figure 4.18. SSM evaluation before and after optimization

struction. The specificity error measures the ability of SSM to only represent instance that belongs to the shape class. The computation goes as follows, a large number of randomly generated instances are obtained by SSM and the error is identified as the difference between the random instance and its closest training set instance. The compactness error is the sum of eigenvalues of variations that measure how compact the SSM is in the shape space. It is seen from Figure 4.18 that these three errors have all decreased from initial correspondence to the optimized correspondence.

Here some comments on landmark number n_P and the number of shapes n_S in the real training set data situation. A practical rule for choosing the landmark number is that the landmarks should be a sufficient representation of the original shape meaning that none of geometric features is left out. One way to quantify this condition is check the relative distance error bound to the original shape. As for shape number, due to the limited data source, the training set in this application comes from only six patients. This small population is in general not statistically meaningful in the sense that the shape modes have limited use; however it is more important to establish that our proposed method can be successfully applied to shapes with certain topological complexities commonly see in the medical fields. It is believed that the proposed algorithm can give more reliable modes if there are more shape instances coming into our training set pool in the future.

CHAPTER 5
APPLICATION TO PROXIMAL FEMUR

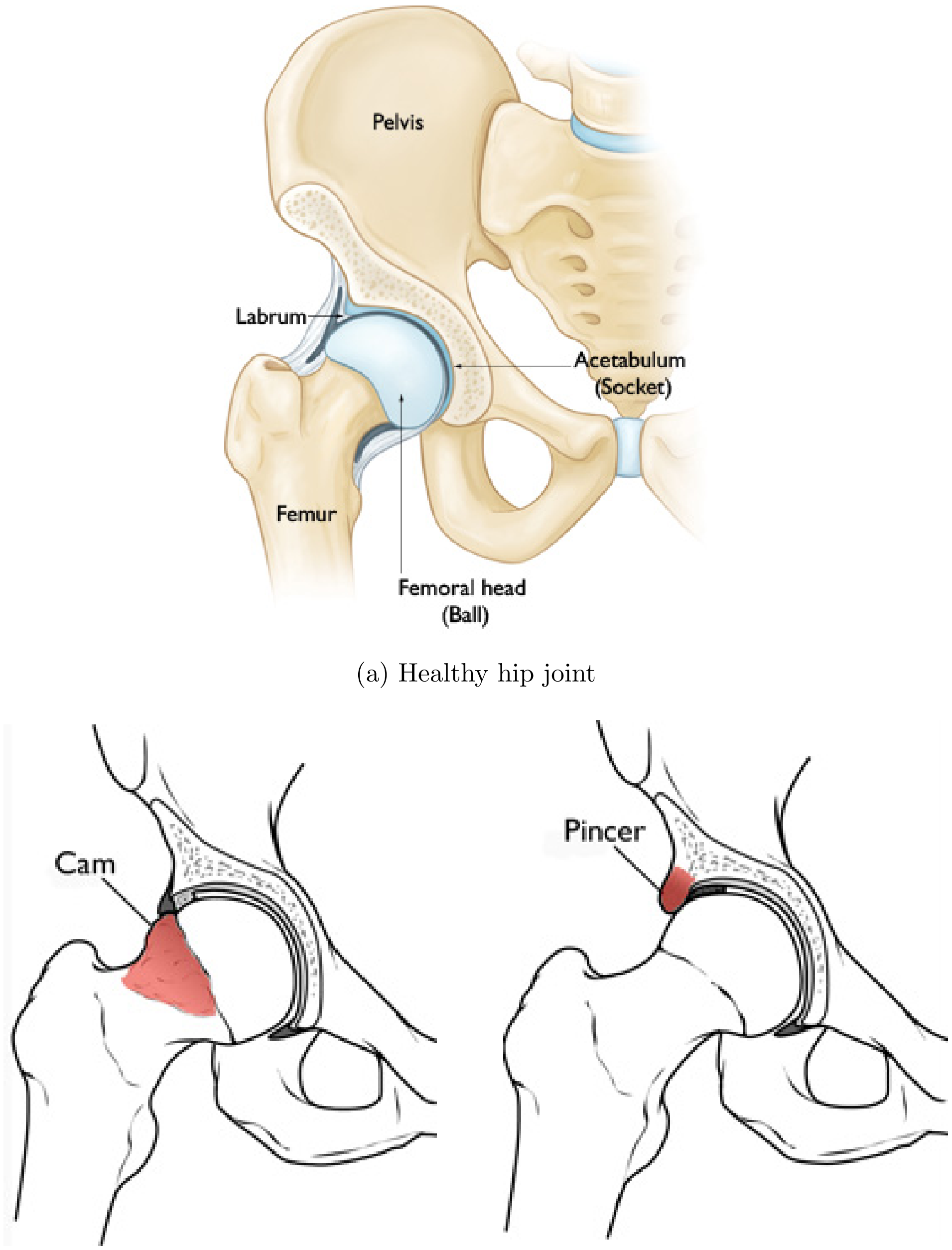
5.1 Introduction

Femoroacetabular impingement (FAI), is a syndrome when the femoral head ball runs abnormally or prohibits a normal range of motion in the acetabular socket. This happens due to ill-shaped hip bones, i.e. the proximal part of femur bone and/or the acetabular socket bone on pelvis. As they do not fit together perfectly, the rubbing against each other causes damage to the joint. Figure 5.1(a) displays the healthy ball-and-socket anatomical structure at the hip joint, where no impingement occurs. Depending on which side of the joint develops shape abnormality, there are three types of FAI:

- *Cam.* Occurs when extra bone extends along the upper surface of the femoral head ball, which is not round and cannot rotate smoothly inside the acaetabulum socket, as seen in Figure 5.1(b). This type of impingement leads to cartilage lesion inside the acetabulum.
- *Pincer.* Occurs when excessive bone material grows beyond the normal rim of the acetabulum cup as highlighted in red in Figure 5.1(c). This type of impingement often crushes the labrum.
- *Combined.* Mixture of both.

As a preliminary study, this thesis will only focus on the cam type impingement and apply the proposed method to a data set of proximal femurs, and the influence on the acetabular side of the joint is ignored for the moment.

5.2 Training set data preparation



(a) Healthy hip joint

(b) Cam impingement

(c) Pincer impingement

Figure 5.1. Femoroacetabular impingement (FAI). Courtesy of American Academy of Orthopaedic Society (AAOS)

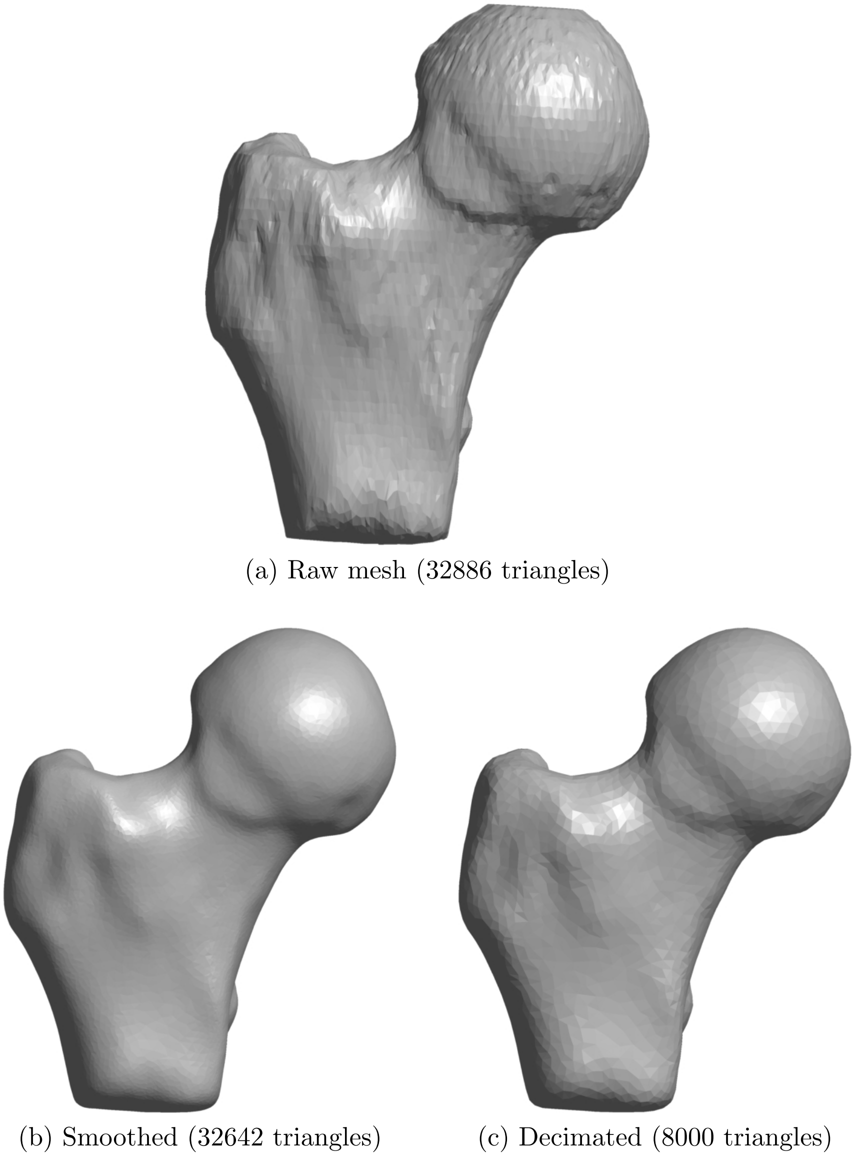
5.2.1 Raw data denoising, smoothing and decimation. Thanks to Rush University Medical Center, 29 proximal femur bone in the form of triangle mesh have been collected for statistical analysis using our proposed approach. The 29 proximal femur bones come from 17 patients, and there are 13 healthy bones and 16 unhealthy. The health status label is determined by whether the patients are experiencing a hip joint pain in testing joint movements.

Table 5.1 lists a comprehensive set of meta data information of all proximal femur meshes. Each shape has an original ".STL" file name and assigned a shape ID. The prefix "HIP" stands for hip joint femoral data, the two digits that follows denotes the patient number, where each patient provides femur bone data on one side or both sides. The first letter after hyphen marks healthy status: "F" means unhealthy when hip joint pain exists (Bad health) and "N" mans "healthy" without pain (Good health). The following letter reveals the side the femur and it is either a "L" (left) femur or "R" (right) femur. These information annotations form the first five columns in the table. The last two columns indicates the number of vertices and triangles of the proximal meshes. The vertex number ranges from 4274 to 36720.

The raw data meshes contain a good amount of outliers, noise and incomplete data. It's advisable to pre-process the mesh for better mesh quality in downstream procedures. Take one shape as an example, the original raw mesh in Figure 5.2(a) contains a few outlying triangles and un-smooth regions resulting from error during the course of MRI acquisition and mesh reconstruction. The superior part on the femur head contains an artificial platform as a result of misplaced cutting and closing. The mesh defects are resolved by denoising and smoothing and the new mesh is shown in Figure 5.2(b) with greatly enhanced mesh quality. For the purpose of efficient feature to storage ration, the mesh is further simplified to the one shown in Figure 5.2(c).

Table 5.1. 29 proximal meshes information: Shape ID, STL file name, Patient ID, Side, Health, Vertex Number, Triangle Number. 29 meshes from 17 patients; 13 healthy, 16 unhealthy

ID	STL	Patient	Side	Health	Vertex Number	Triangle Number
1	HIP01-FRF	A(01)	Right	Bad	14949	29890
2	HIP01-NLF	A(01)	Left	Good	13876	27724
3	HIP02-FRF	B(02)	Right	Bad	16459	32886
4	HIP02-NLF	B(02)	Left	Good	17259	34490
5	HIP04-FLF	C(04)	Left	Bad	4376	8748
6	HIP04-NRF	C(04)	Right	Good	4274	8544
7	HIP05-FLF	D(05)	Left	Bad	36720	73436
8	HIP07-FRF	E(07)	Right	Bad	29826	59648
9	HIP07-NLF	E(07)	Left	Good	28316	56628
10	HIP10-FRF	F(10)	Right	Bad	16730	33456
11	HIP10-NLF	F(10)	Left	Good	16508	33012
12	HIP11-FRF	G(11)	Right	Bad	16854	33700
13	HIP11-NLF	G(11)	Left	Good	16732	33452
14	HIP13-FLF	H(13)	Left	Bad	20121	40234
15	HIP13-NRF	H(13)	Right	Good	20687	41358
16	HIP14-FRF	I(14)	Right	Bad	25052	50100
17	HIP14-NLF	I(14)	Left	Good	25098	50192
18	HIP15-FLF	J(15)	Left	Bad	22538	45072
19	HIP15-NRF	J(15)	Right	Good	21944	43884
20	HIP16-NLF	K(16)	Left	Good	13929	27854
21	HIP21-FRF	L(21)	Right	Bad	6595	13186
22	HIP21-NLF	L(21)	Left	Good	6500	12996
23	HIP22-FLF	M(22)	Left	Bad	34516	69028
24	HIP23-FRF	N(23)	Right	Bad	9288	18568
25	HIP23-NLF	N(23)	Left	Good	9056	18018
26	HIP24-FLF	O(24)	Left	Bad	31362	62720
27	HIP26-FLF	P(26)	Left	Bad	12522	25040
28	HIP26-NRF	P(26)	Right	Good	12906	25808
29	HIP27-FRF	Q(27)	Right	Bad	6530	13048



(a) Raw mesh (32886 triangles)

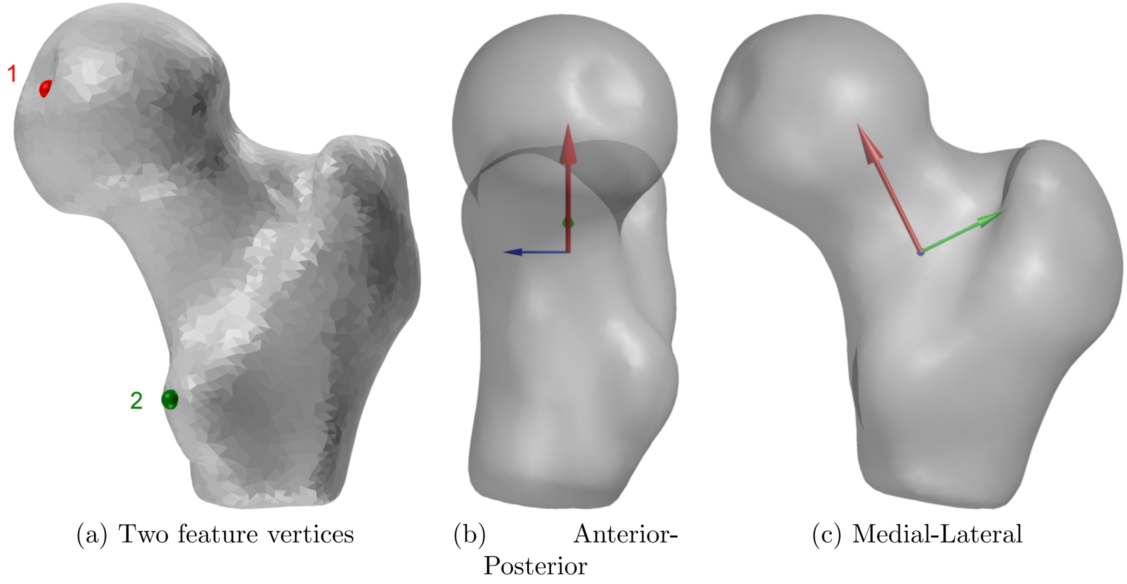
(b) Smoothed (32642 triangles)

(c) Decimated (8000 triangles)

Figure 5.2. Data denoising, smoothing and decimation

5.2.2 Feature identification, orientation and mesh cutting.

Next two feature



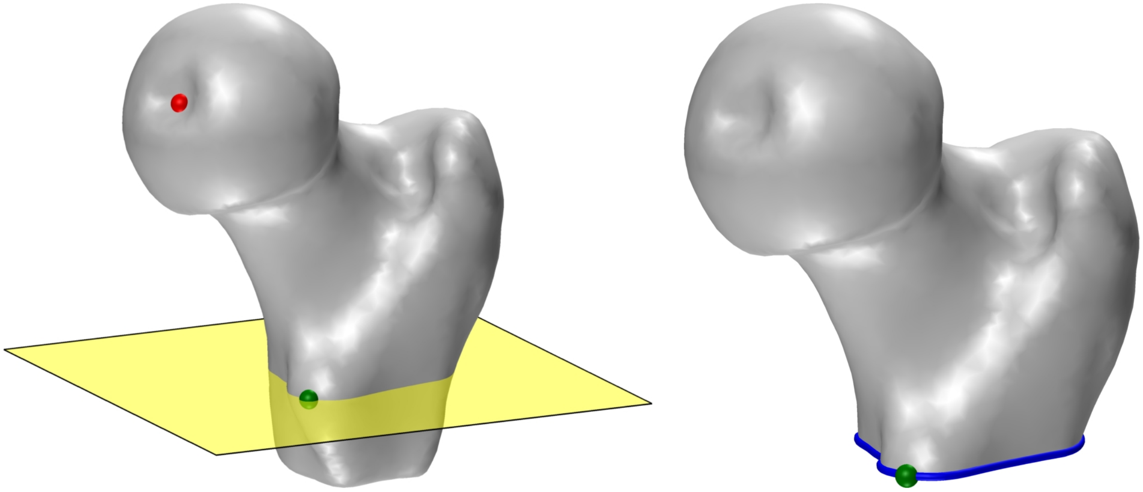
(a) Two feature vertices (b) Anterior-Posterior (c) Medial-Lateral

Figure 5.3. Two feature vertices identified (fovea in red, less trochanter eminence in blue) and proximal femur orientation

vertices are marked on each shape mesh as geometric guide for subsequent operations, the first feature vertex is located at the fovea and the second feature vertex at less trochanter eminence. Figure 5.3(a) shows the locations of the two feature vertices on one example shape.

Since the raw data is closed mesh and our algorithm works on geometry that is topologically equivalent to a disk, it is necessary to cut the bottom off and form an open mesh to make the proximal femur topologically compatible. The planar cut is chosen as a simple viable way. As the cutting plane requires an orientation, it is important that all the femurs should be reoriented consistently across the entire training set of shapes. The non axis aligned bounding box [81] of all the mesh vertex points becomes useful in this situation. In the first step, the principal axis of the obtained bounding box with shortest span (the third axis shown in blue in Figure 5.3(b)) is aligned with the anterior-posterior direction. Then in the second step the whole femur (including the unshown femur shaft and distal portion) axis is manually decided to be the gray dotted line in Figure 5.3(c), which is aligned with the superior direction. The first step is done automatically and the second step is semiautomatic due to the involved manual axis determination process. After this reorientation, the femur meshes are brought to a stance that is close enough to its natural orientation on the patient's body.

Recall from Table 5.1 that there are 13 right femurs and 16 left femurs, in order for the ease of correspondence discussion, we convert all left proximal femurs into right ones by flipping them about the sagittal plane so that all the proximal femurs are right femur.



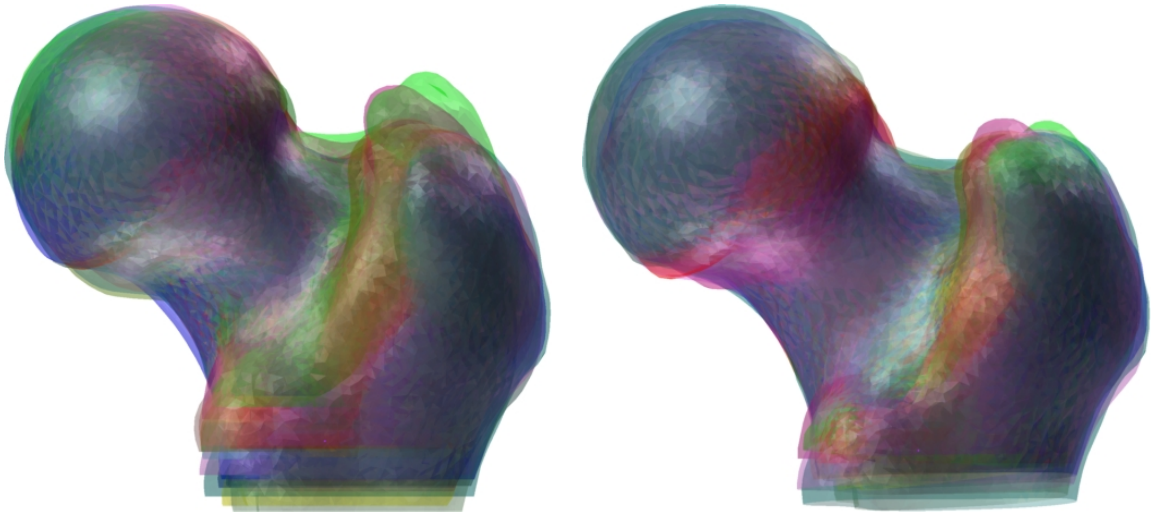
(a) Cutting plane through feature point 2

(b) Mesh after planar cut

Figure 5.4. Proximal femur mesh with planar cut

After the superior orientation has been established, the mesh is cut by a planar whose normal aligns with superior direction (facing up) and that passes through feature point 2 in green (less trochanter eminence) as seen in Figure 5.4(a). The remaining mesh in Figure 5.4(b) with boundary in purple is an open mesh of interest on which our proposed method will be run on.

5.2.3 ICP alignment. With all the meshes cut into open mesh, the Iterative Closest Point (ICP) algorithm [82] is employed to bring them into better alignment.



(a) Before alignment

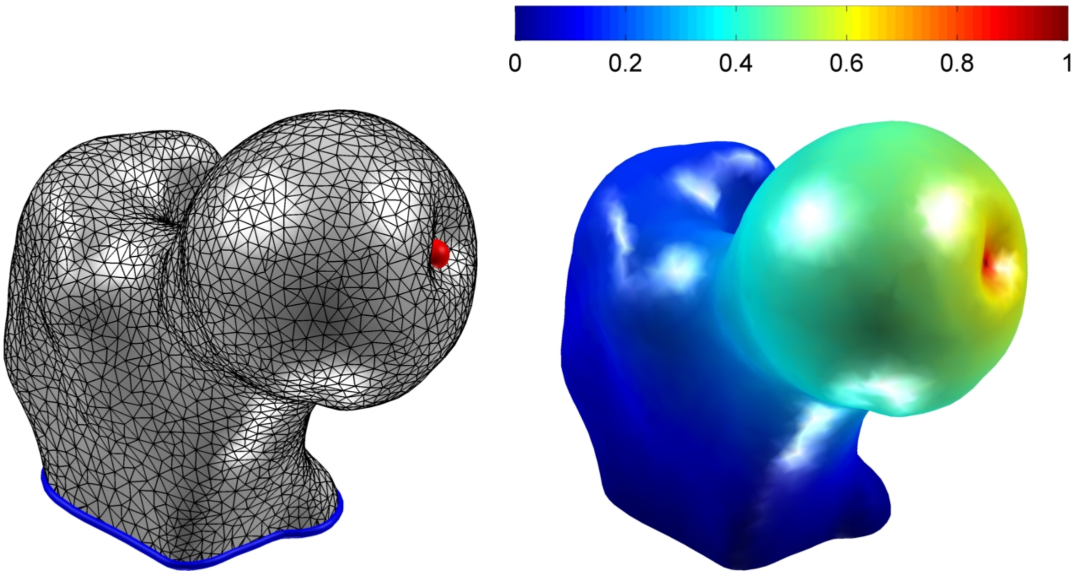
(b) After alignment

Figure 5.5. Mesh alignment by Iterative Closest Point algorithm

Figure 5.5(a) and (b) show the 29 femur meshes before and after the ICP alignment.

5.2.4 B-spline fitting. Since our proposed direct reparametrization method takes a training set of B-spline geometry as input, what needs to be done is the B-spline fitting of the proximal meshes. As B-spline inherently implies a parametrization, the B-spline fitting often starts with a mesh parametrization operation that maps the mesh to a regularly shaped parameter domain. The topic of mesh parametrization is extensively and deeply discussed in the graphics community [57][33]. The current open mesh of proximal femur has a topology equivalent to a disk and can be mapped to a square parameter domain directly, where the mesh boundary corresponds to the four square sides. However, this mapping option is unable to the control the parametrization quality in the interior, and the resultant B-spline suffers from severe creases in the femur neck area. This is because that the iso-parametric lines the two parametric directions U and V are meeting at a small angle far away from 90°. In the square parameter domain, the iso-parametric lines along U and along V are perpendicular everywhere; in the physical domain, a larger deviation from right angle of the meeting between iso-parametric U lines and iso-parametric V lines, a larger angle distortion is introduced, making the fitted B-spline more likely to be in poor quality.

Solve harmonic V field. In order to monitor the parametrization quality that direct affects B-spline surface quality, the harmonic field method [83] is utilized to compute a high quality mesh parametrization in the sense that iso-parametric lines along U and V directions meet perpendicularly nearly everywhere on the mesh. At the first stage, a harmonic V-field is to be calculated by solving a discrete Laplace equation that is constrained on the mesh surface and subject to Dirichlet boundary conditions (BC). Specifically as displayed in Figure 5.6(a), the pre-selected fovea vertex (red) has $V = 1$ and all the boundary vertices (blue) has $V = 0$. The discrete Laplace operator is equivalent to setting each internal vertex's V-value to the weighted average of all its neighbors' V-values. Doing this for all internal vertices leads to a linear system of equations, and the selection of averaging weights has influence on the parametrization result. As recommended by [83] [84], here the *Discrete Harmonic* weights are chosen to find the V-field shown in Figure 5.6(b). The result can be regarded as a steady



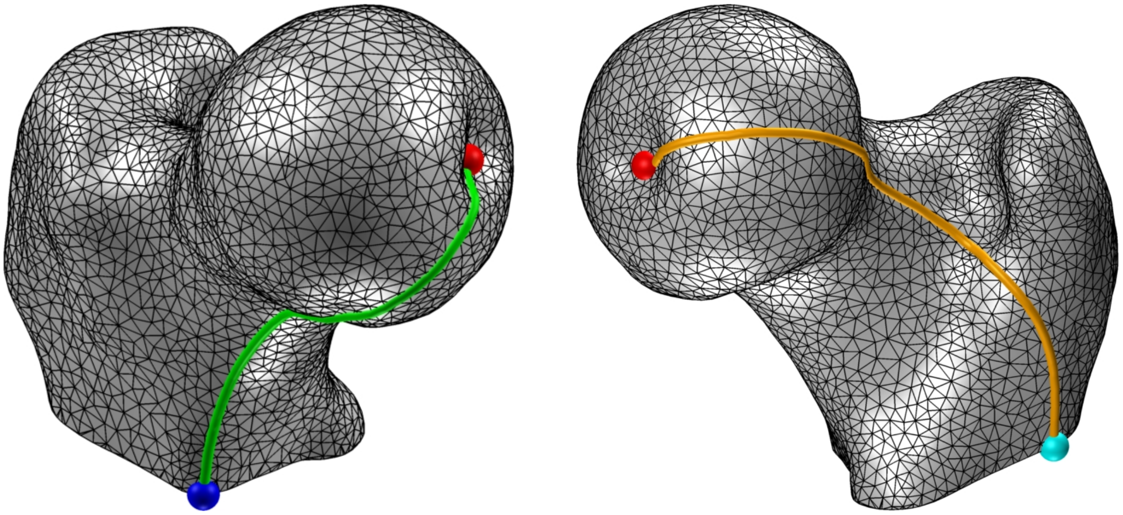
(a) Boundary condition

(b) Harmonic V field

Figure 5.6. Mesh alignment by Iterative Closest Point algorithm

state temperature field after a heat source is placed at fovea vertex.

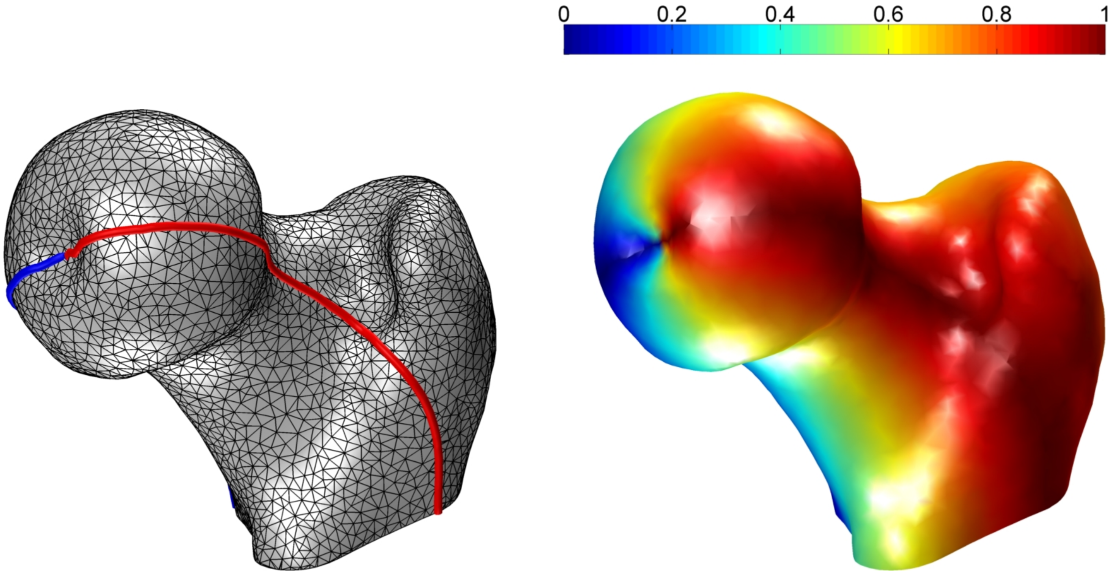
Solve harmonic U field. After obtaining V-field, a vertex at boundary is selected as the anterior-medial corner, and will be referred to as “generatrix start”. It is shown in blue in Figure 5.7(a). A gradient line is then traced starting from the generatrix start and along the gradient direction of the V-field ∇V. The computation of ∇V with a mesh triangle with harmonic field V is described in [83]. The traced line will be called “generatrix” and is colored in green in Figure 5.7(a). Another boundary point, cyan in Figure 5.7(b), can be picked at halfway of generatrix start, and it is called “opposite generatrix start”. Tracing from this point produces another gradient line, which is shown in orange and called “opposite generatrix”. The naming comes from the intension of mapping the proximal mesh onto a cone, where the fovea corresponds to the apex and the generatrix gradient line the generatrix.


(a) Generatrix tracing along ∇V

(b) Opposite generatrix tracing along ∇V

Figure 5.7. Mesh alignment by Iterative Closest Point algorithm

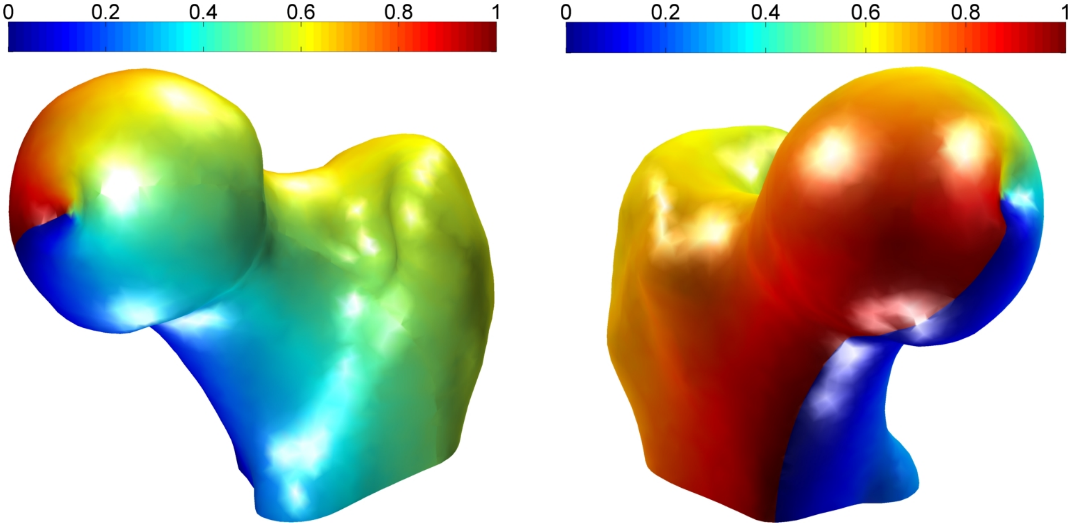
These two traced gradient lines are then added into the mesh and will serve as the boundary condition of solving the harmonic U-field. It is seen in Figure 5.8(a) that the two gradient lines divide the mesh into two regions. With each region, the regional harmonic U-field can be solved by the harmonic field method described earlier



(a) Boundary conditions (b) Two regionsl harmonic U-fields

Figure 5.8. Regional harmonic U-field

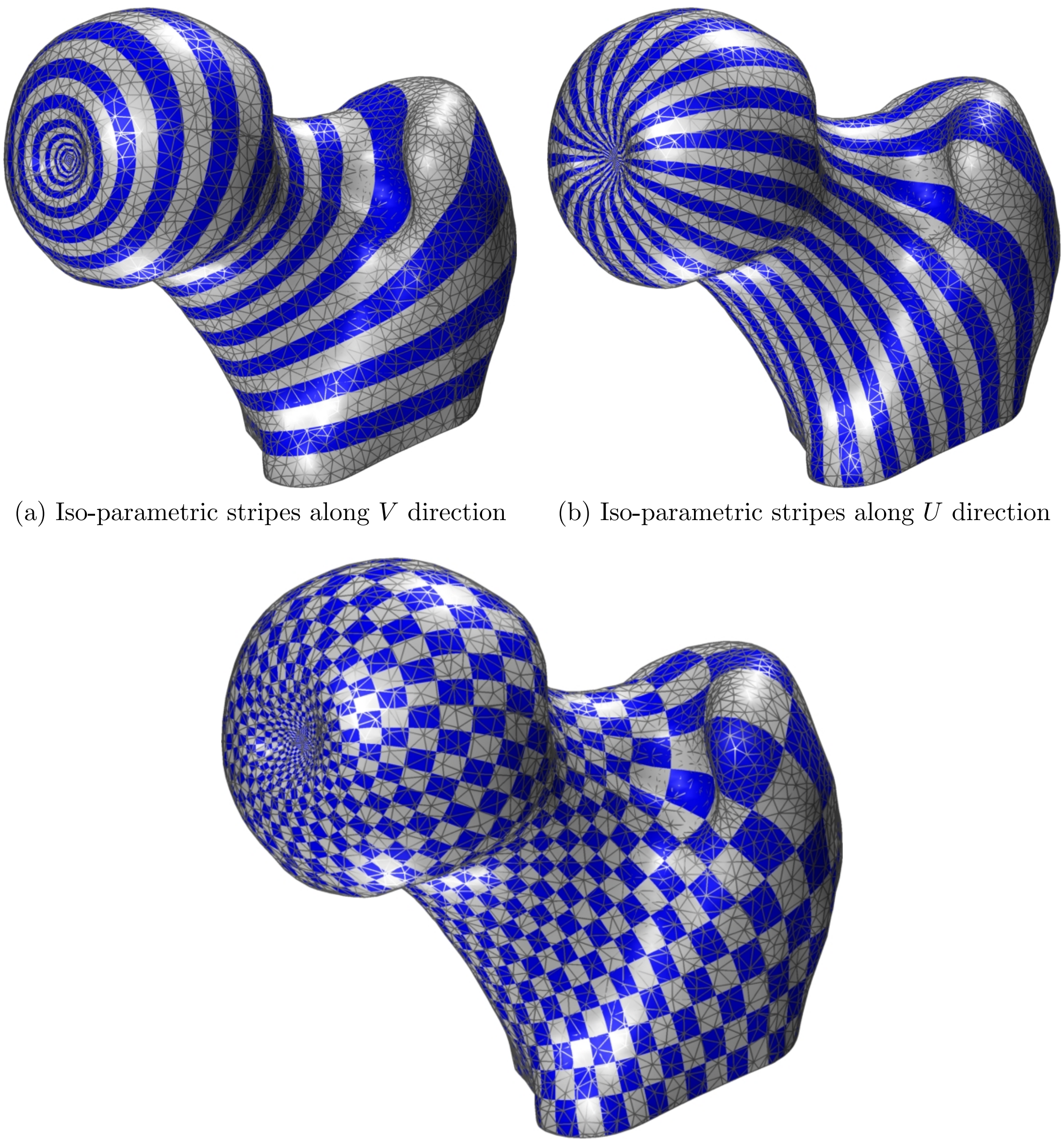
where the Dirichlet boundary conditions are both: $U = 0$ on the generatrix (blue), and $U = 0.5$. on the opposite generatrix (red). The regional harmonic U-fields are solved and shown in Figure 5.8(b).



(a) Posterior viewpoint (b) Anterior viewpoint

Figure 5.9. Harmonic U-field after reconciliation

The two regional U-fields can be reconciled to form a consistent harmonic U-field by keeping one unchanged and scaling the other. After reconciliation, the whole harmonic U-field is shown in Figure 5.9(a) where the discontinuity at $U = 0.5$ of opposite generatrix disappears. The only discontinuity remains at $U = 0$ (also U=1) at the generatrix as observed in Figure 5.9(b).



(a) Iso-parametric stripes along V direction

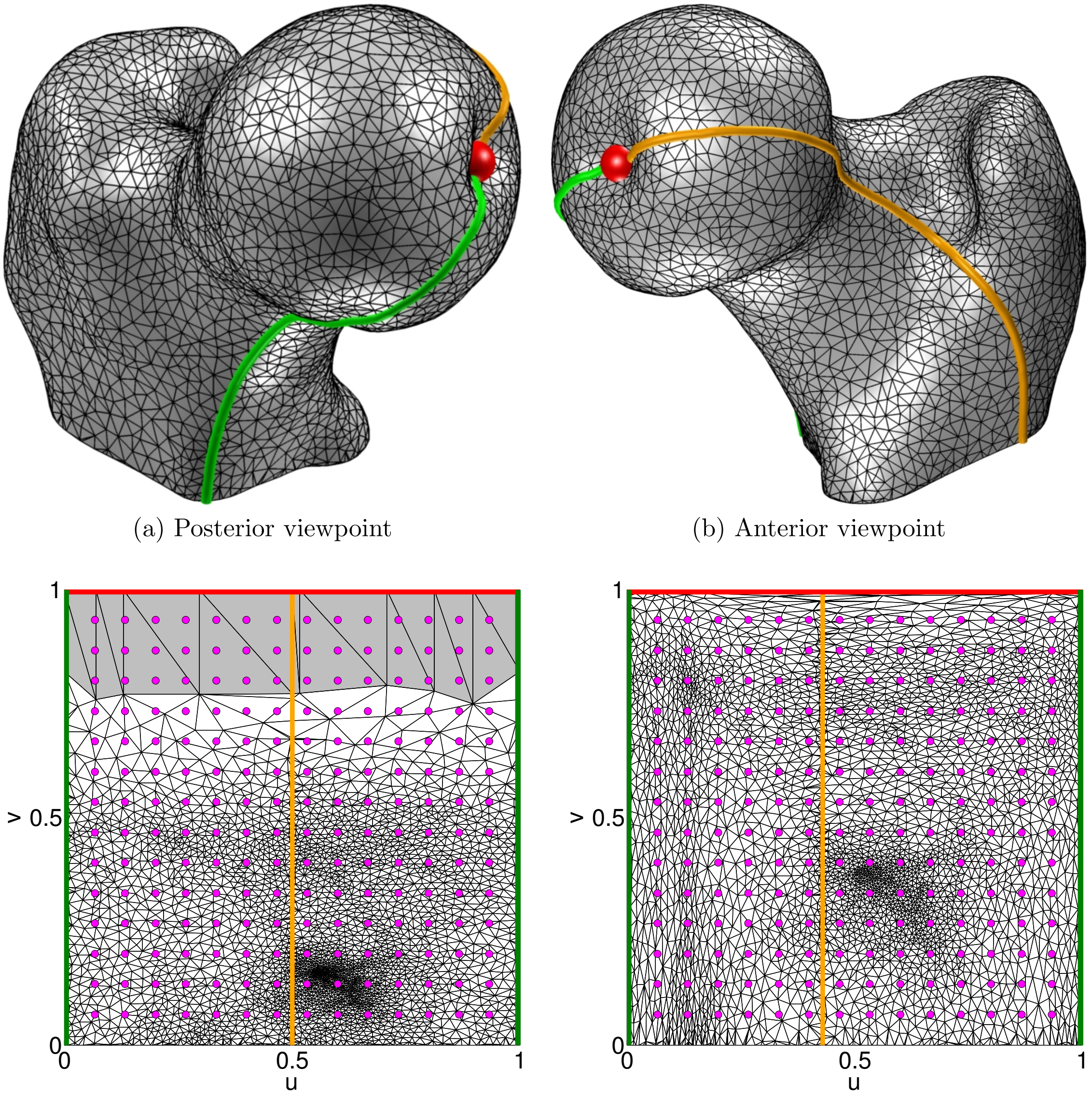
(b) Iso-parametric stripes along U direction

(c) Iso-parametric checkerboard

Figure 5.10. Iso-parametric lines and mesh parametrization

Now every vertex, except fovea and generatrix vertices have unique U- and V-values, which direct leads to a mapping onto the parameter domain. The isoparametric lines along U and V directions are displayed in stripes plotted in Figure 5.10(a) and (b). The overlaid stripes shows the parametrization checkerboard, which the isoparametric lines in the two parametric directions meet at right angle everywhere on the mesh except near fovea vertex. This suggests that a conformal mapping has been established that preserves the angle.

Mesh parametrization improvement. The parametrization just achieved



(a) Posterior viewpoint

(b) Anterior viewpoint

(c) Posterior viewpoint

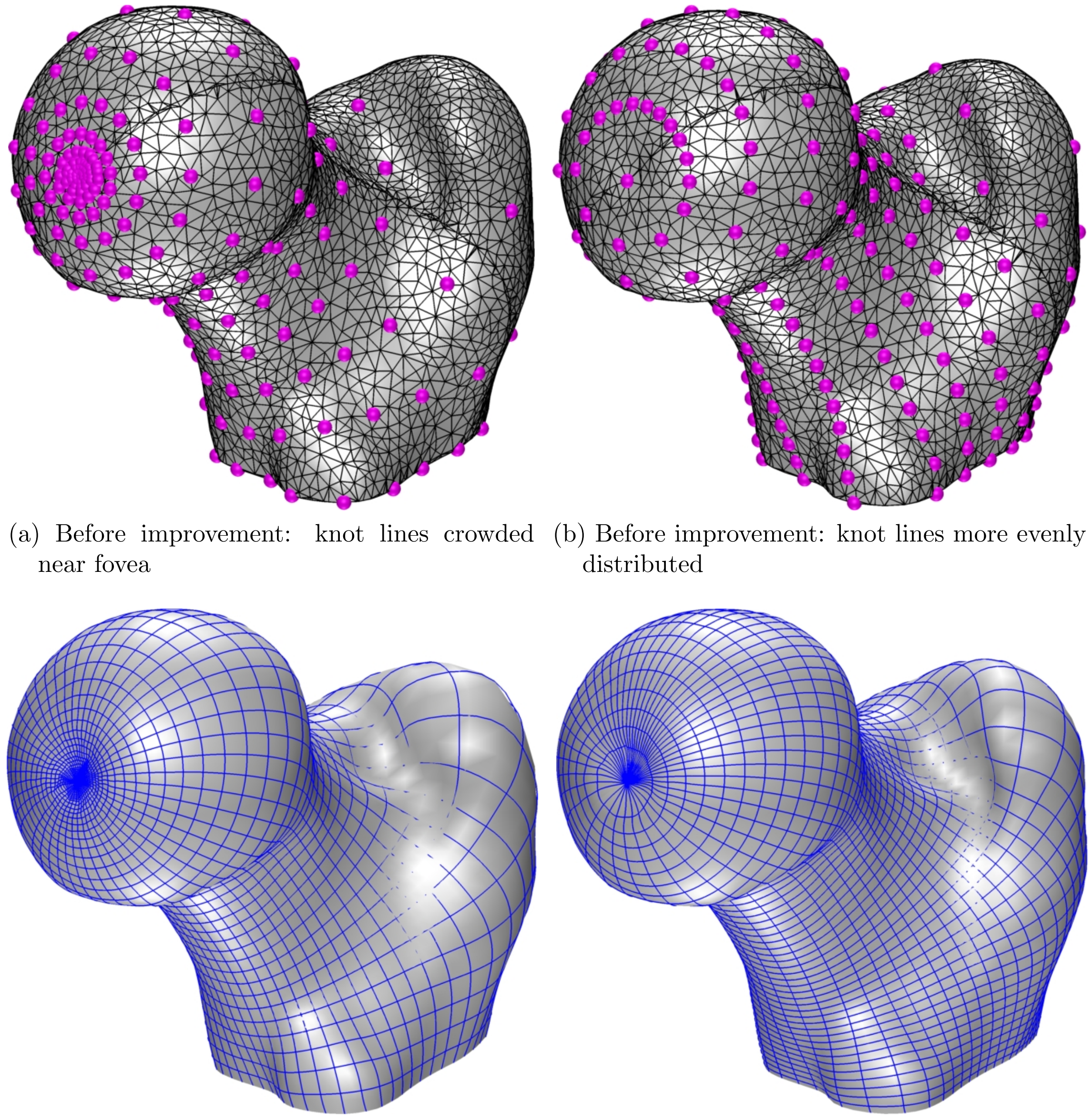
(d) Anterior viewpoint

Figure 5.11. Mesh parametrization improvement

has built a mapping from the the proximal femur mesh in the physical domain to the square parameter domain. What lies in between in the mapping is an intermediate cone domain, and it can be thought of as being formed by joining the two sides of the square and collapsing the top side. In order for a better visualization of this parametrization from the mesh through cone domain to square domain, Figure 5.11(a) and (b) show the fovea in red (cone apex), generatrix in green (cone generatrix) and opposite generatrix in orange. These three entities will be mapped to the top side (red), left and right sides (green) and middle line (orange) respectively in the square domain as seen in Figure 5.11(c). Notice that the left and right sides (green) correspond to the same generatrix gradient line in physical domain, and entire the top side (red) vertices in the square parameter domain all correspond to only one vertex, i.e. the fovea vertex. The gray triangles signifies all incident triangles to the fovea vertices.

Regular sampling and B-spline fitting. Next a regular grid of sampling parameters denoted by the purple dots in Figure 5.11(c) is used to generate a counterpart in the physical domain as seen in Figure 5.12(a). The grid shown in of resolution 15 × 15, and a regular grid of resolution 200 × 200 is actually used to serve as the fitting data. The B-spline in Figure 5.12(c) is fitted from the data by minimizing the least square with respect to control positions according to the techniques prescribed in [32].

Figure 5.11(c) reveals one problem that quite a large portion of the sampling parameters are within the triangles incident to the fovea vertex; this leads to oversampling in this reliable area and undersampling in other more important areas such femur neck and trochanter etc. This unreasonable sampling pattern is caused by the existing mesh parametrization shown in Figure 5.11(c), where the small number of the gray shaded triangles in the square parameter domain account for almost 20% of the entire domain area.



(a) Before improvement: knot lines crowded near fovea

(b) Before improvement: knot lines more evenly distributed

(c) Before improvement: knot lines crowded near fovea

(d) Before improvement: knot lines more evenly distributed

Figure 5.12. Fitted B-spline improvement

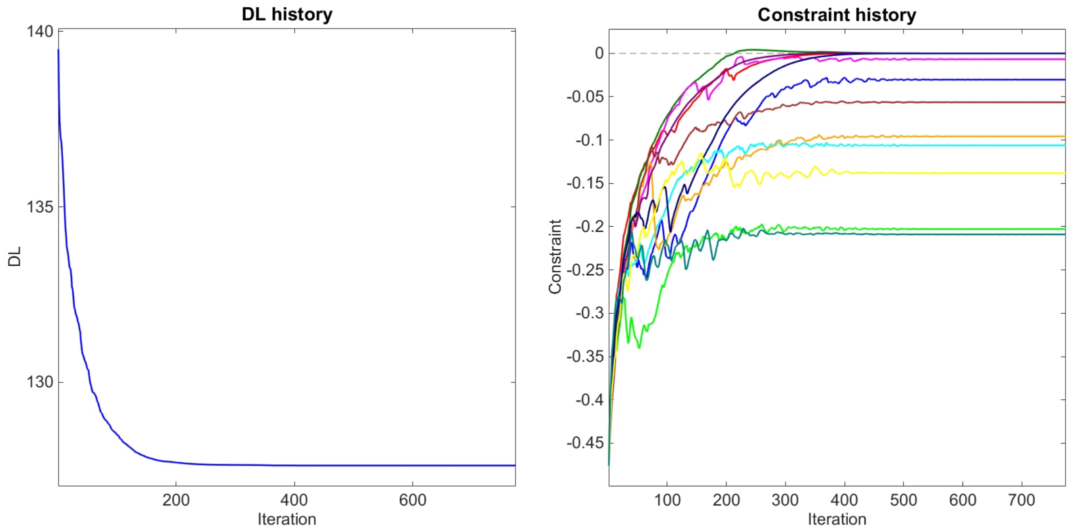
For the purpose of alleviating this issue, one viable solution is to adjust the vertices along only the U and V direction so as to preserve the connectivity. The improvement can be done by an optimization problem which minimize the mesh stretch [58] with a reparametrization B-spline coefficients at the two directions as design variables. After optimization, the mesh parametrization sees remarkable improvement in Figure 5.11(d). The improved parametrization in turn leads to a more reasonable sampling distribution in the physical domain as shown in Figure 5.11(d). Ultimately the fitted B-spline is much better parametrized as exhibited by its knot line patterns, where there are far fewer knot lines near fovea and more in other important areas after parametrization improvement. All the 29 proximal femurs will be processed with this routine for the preparation of B-spline training data.

5.3 Correspondence optimization and statistical analysis

5.3.1 Correspondence optimization and results. The 29 proximal B-spline surfaces are divided into two groups according to their health status, one is healthy group of 13 B-spline surfaces, and the other unhealthy group with 16 B-splines. The proposed algorithm in previous chapters are used within each group and two statistical models can be obtained accordingly. For both groups, 51×51 landmarks are used and 9×9 reparametrization B-spline are employed.

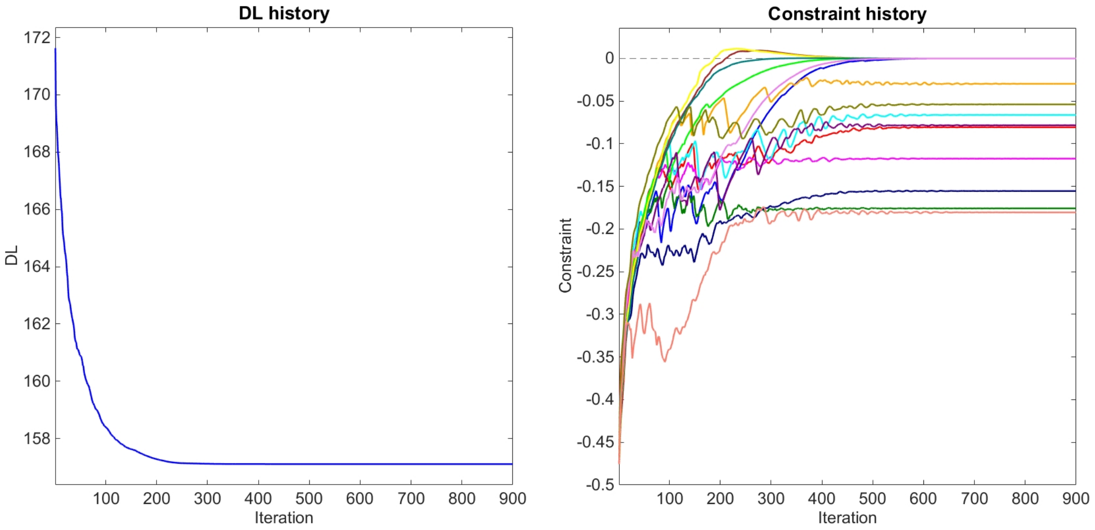
5.3.1.1 Optimization history. The optimization history of objective function (description length) and aggregated constraint of Jacobian are recorded and plotted in Figure 5.13(a) and (b) respectively for the healthy group. Similar plotting is done for unhealthy group as well and the history is shown in Figure 5.14. It is seen that the objective function value has dropped after optimization and some of the diffeomorphism constraints have become active at the ending optimized iteration.

5.3.1.2 Statistical model evaluation measures improvement. Again the



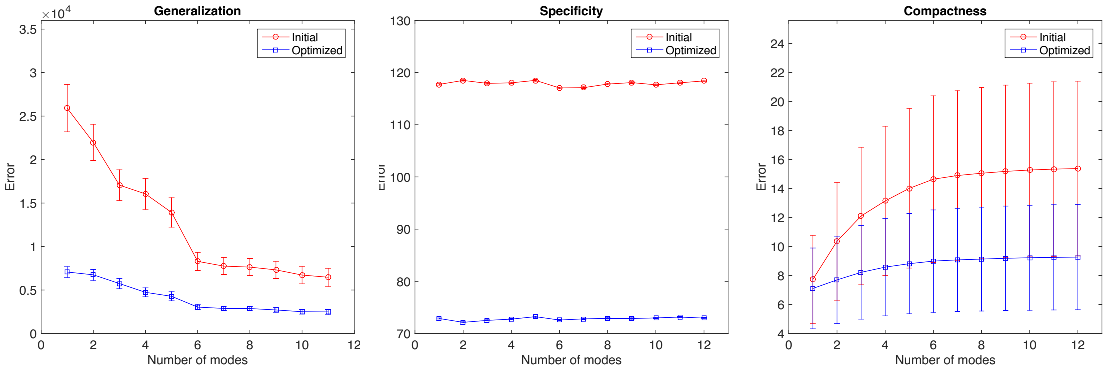
(a) DL history (b) Constraint history

Figure 5.13. Correspondence optimization history for HEALTHY group



(a) DL history (b) Constraint history

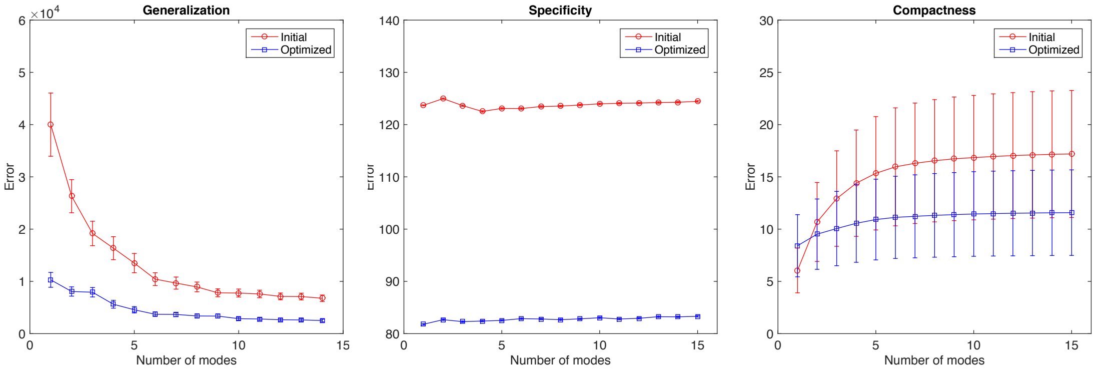
Figure 5.14. Correspondence optimization history for UNHEALTHY group



(a) Generalization ability (b) Specificity (c) Compactness

Figure 5.15. SSM evaluation improvement for HEALTHY group

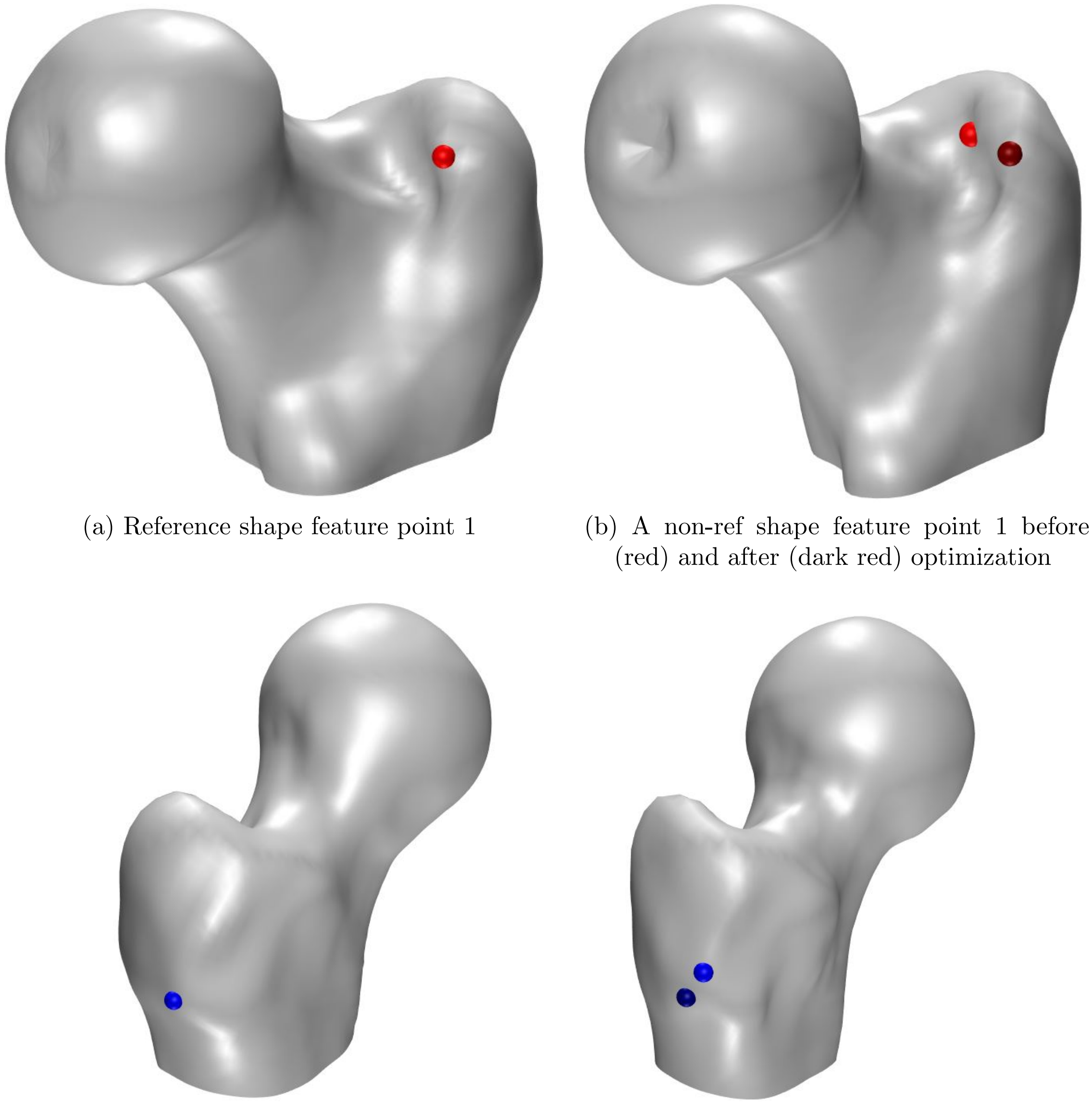
optimization results have been tested for improvement in terms of the three statistical model evaluation measures proposed in [28]. The generalization ability, specificity and compactness measures have all seen improvements after performing our proposed optimization algorithm as observed in Figure 5.15 for healthy group and Figure 5.16 for unhealthy group.



(a) Generalization ability (b) Specificity (c) Compactness

Figure 5.16. SSM evaluation improvement for UNHEALTHY group

5.3.1.3 Correspondence improvement. In addition to SSM evaluation measure improvement, the correspondence across shapes has improved as well. Figure 5.17(a) shows a feature point (trochanter eminence) fixed on the reference shape is in red; its corresponding feature point on a non-reference shape before optimization is shown in red as in Figure 5.17(b), the two points are not in a good correspondence.



(a) Reference shape feature point 1

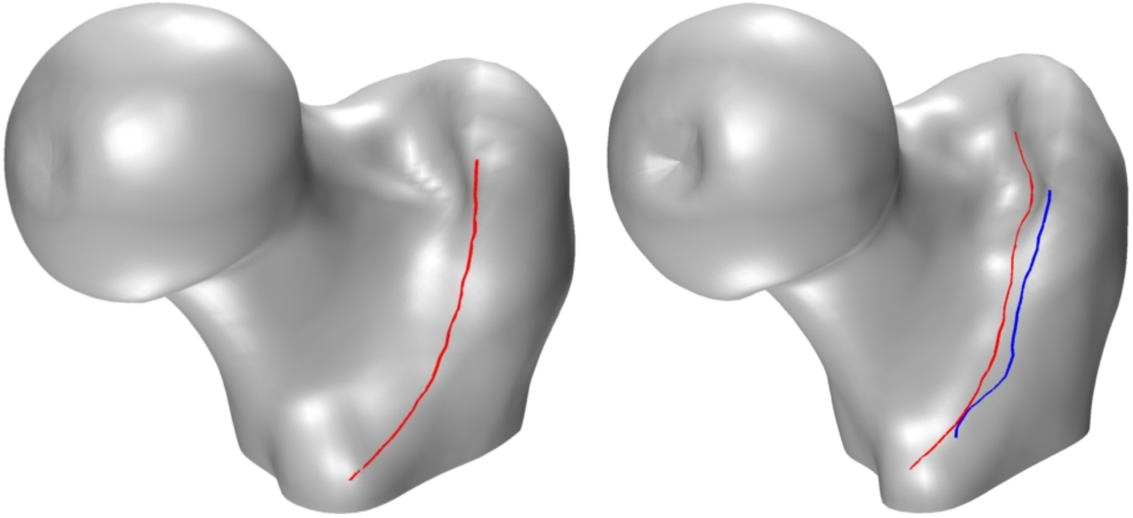
(b) A non-ref shape feature point 1 before (red) and after (dark red) optimization

(c) Reference shape feature point 2

(d) A non-ref shape feature point 2 before (red) and after (dark red) optimization

Figure 5.17. Point correspondence improvement on HEALTHY group

After optimization, the corresponding point of feature point moved to dark red point, which indicates a better correspondence as it moves closer to the trochanter eminence on the non-reference shape. Another feature point in blue as shown in Figure 5.17(c) and (d), where similar correspondence improvements are experienced.



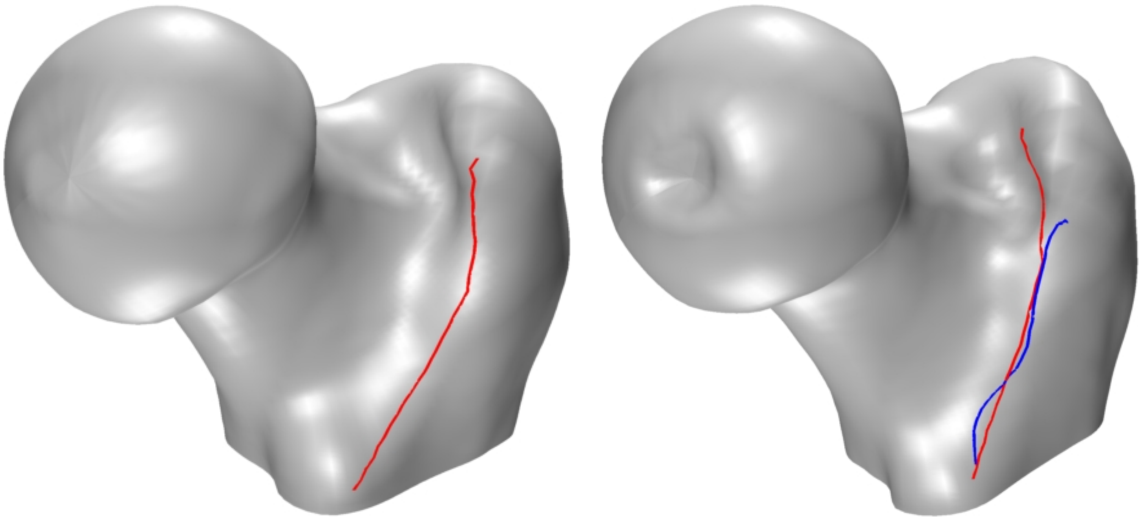
(a) Reference shape feature point 1

(b) A non-ref shape feature point 1 before (red) and after (blue) optimization

Figure 5.18. Feature line correspondence improvement on HEALTHY group

To further explore the correspondence improvement, we identify a feature line on shapes by connecting a series of landmark parameter points as shown in Figure 5.18(a), where the feature line is chosen to be the intertrochanteric line. Before optimization, the corresponding line (red) of the feature line of reference shape is not quite the intertrochanteric line; after optimization the location of corresponding line has moved to a position (blue) that better characterizes the same semantic feature line. Similar results are observed for unhealthy group as shown in Figure 5.19.

As stated previously, the quality of group-wise correspondence directly affect the quality of SSM, therefore here it is necessary to analyze the improvement of correspondence so that the statistical model obtained later faithful reflect the shape variational pattern of the input training set.



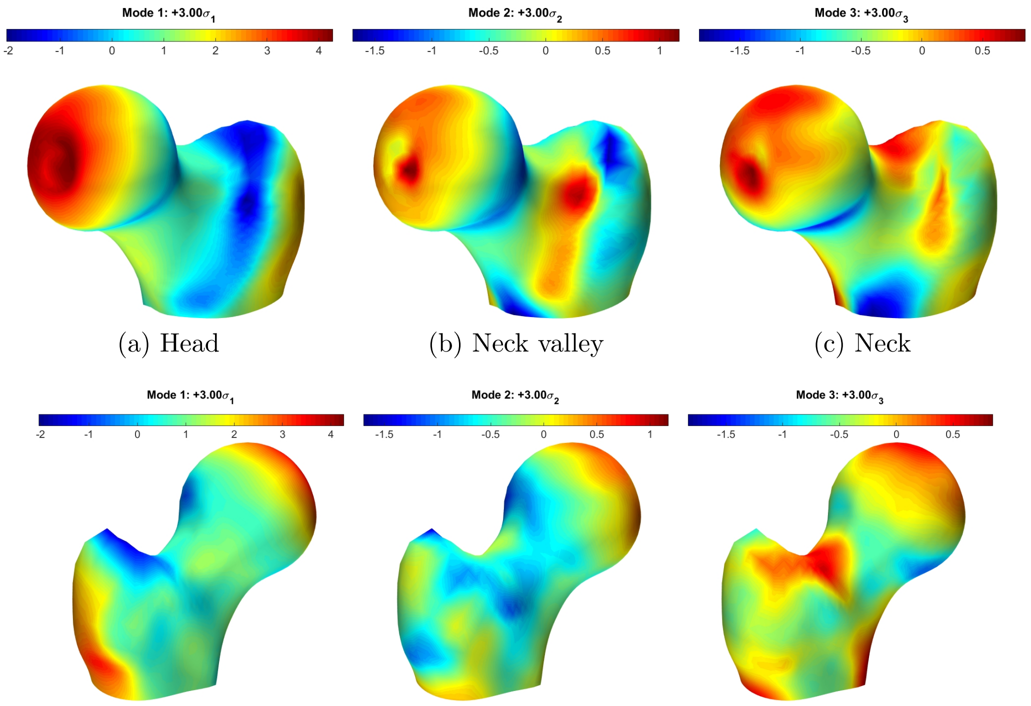
(a) Reference shape feature point 1

(b) A non-ref shape feature point 1 before (red) and after (blue) optimization

Figure 5.19. Feature line correspondence improvement on UNHEALTHY group

5.3.2 Statistical modes and analysis.

5.3.3 Statistlca model and modes. The statistical model, i.e. the mean shape



(a) Head

(b) Neck valley

(c) Neck

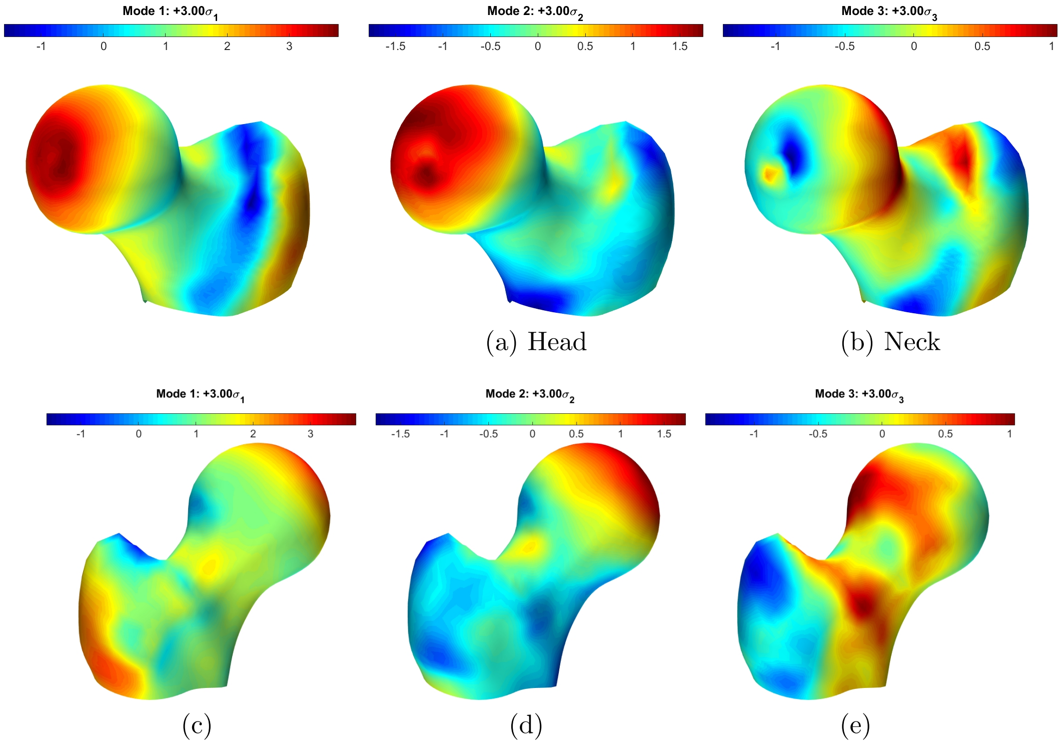
(d)

(e)

(f)

Figure 5.20. Healthy group: first three modes on top of mean shape

and modes, are immediately available once the optimized group-wise correspondence has been found.Figure 5.20 shows the first three modes of healthy group, where the mean shape is compared with the $+3\sigma_m$ mode shape for $m = 1, 2, 3$, and in both anterior and posterior views. The color field represents the shell distance from the mean shape to the mode shapes along the vertex normals on the mean shape, it reflects the shortest moving pattern of a particular mode. The first mode of appears to capture the femur head ball enlarging and the trochanter shrinking variation at the magnitude of about 4.3 mm. The second mode captures the additional thickening pattern in the intertrochanteric valley with a magnitude of 1.2 mm. The third mode reveals a bone material depositing pattern near the femur neck at a magnitude of 0.7mm.



(a) Head

(b) Neck

(c)

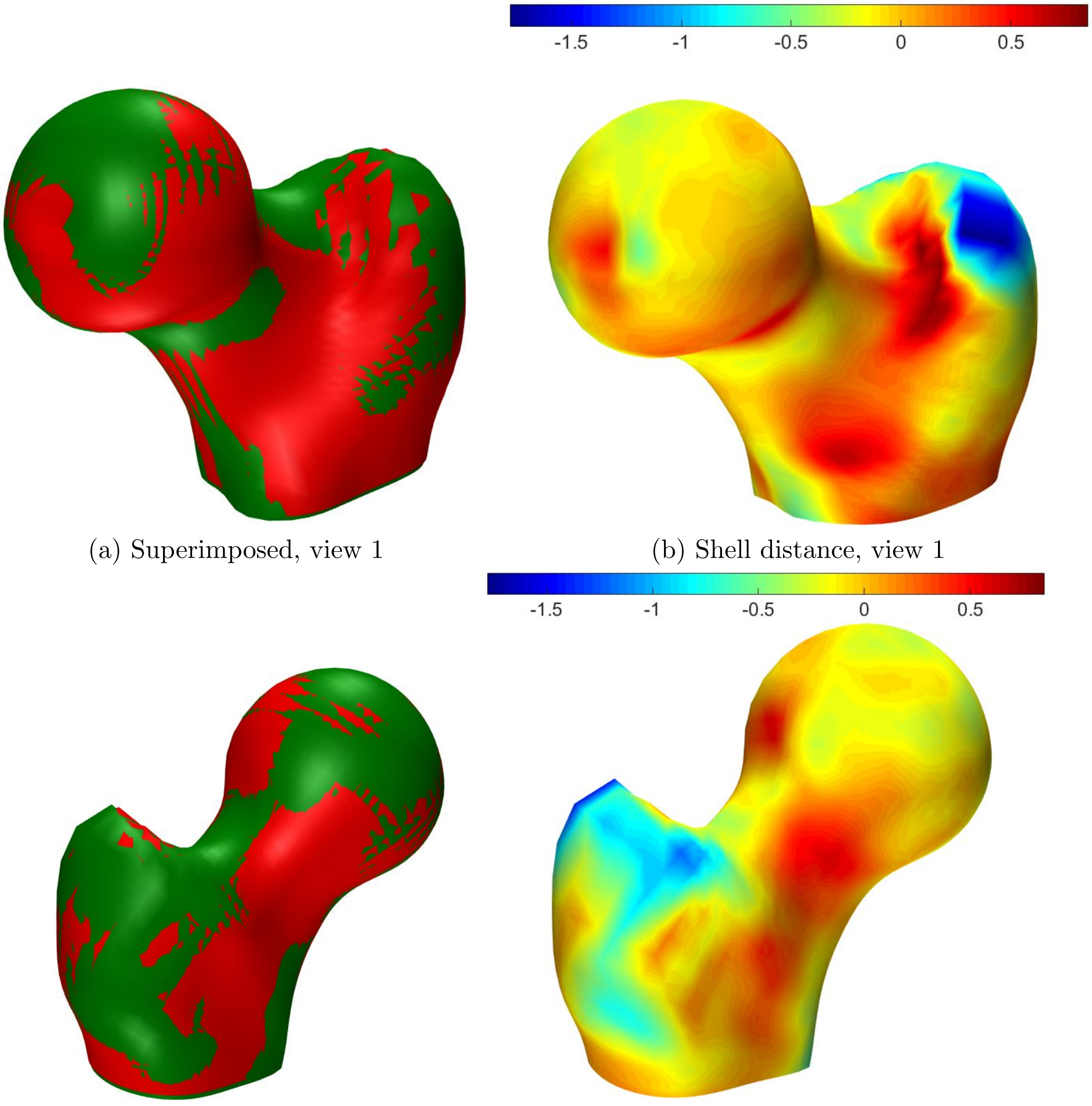
(d)

(e)

Figure 5.21. Unhealthy group: first three modes on top of mean shape

Similar observations can be made for unhealthy group as well, where the second mode captures a larger magnitude of 1.7mm for the femur head ball enlarging variation, as compared to that of healthy group. It is worth noticing that in the third mode of unhealthy group, the femur head ball portion closer to the femur neck is demonstrating a clear thickening pattern that suggests the excessive bone material phenomenon across all unhealthy proximal femurs. This is dissimilar to the third mode from healthy group, where only the superior portion of the femur head sees variation and the portion closer the the neck does not.

5.3.4 Healthy mean vs. unhealthy mean. Other than intra group statistical analysis, it is also meaningful to conduct inter-group comparison. Figure 5.22 com-



(a) Superimposed, view 1

(b) Shell distance, view 1

(c) Superimposed, view 2

(d) Shell distance, view 2

Figure 5.22. UNHEALTHY mean on top of HEALTHY mean

pares the mean shape of healthy and unhealthy group by superimposing them in two anatomical views. Figure 5.22(a)(c) displays clearly the bone surface order, which shows at the neck region, the unhealthy mean shape is truly having more material grown, which causes the impingement and the pain reaction typically seen in the unhealthy bones. Figure 5.22(b)(d) offers a quantitative assessment of this comparison, and a magnitude of 0.8mm discrepancy is observed mainly near the femur head ball portion that is closer to femur neck. These inferences based on statistical models provide valuable insights, and it has also been confirmed the clinical conclusions drawn by our collaborators in RUSH hospital.

CHAPTER 6
CONCLUSION

6.1 Summary

This thesis introduces a novel reparameterization method for population-based correspondence optimization for statistical shape modeling. Each shape instance is represented as a differentiable B-spline surface. The reparameterization of each surface is also represented via B-splines. The diffeomorphic reparameterization is cast as constraints on the B-spline coefficients. This resulting large-scale optimization is then solved via a gradient based approach. To facilitate the convergence, full analytical gradients of the cost function (i.e. description length) with respect to manipulation parameters (i.e. B-spline coefficients) have been derived. The adjoint approach for computing the gradient has found to be especially efficient. This B-spline based direct diffeomorphic reparameterization is found to be effective on a set of both synthetic and real data sets. It typically achieves more than one orders of magnitude speed gain than concatenation of simple mapping based reparamaterization methods such as Cauchy kernels and CPS warps for correspondence optimization.

In addition, we have presented methods for accurately and efficiently computing continuous formulations of the covariance matrix where B-splines are used both as a shape representation and as a form of reparameterization. We have shown, with B-spline representation of the shapes, the formulation I is amenable to analytical computing without sampling or discretization. Numerical approaches based on mid-point and Gauss quadrature are developed for approximating both continuous formulations. We have shown 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 shown 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 potential collapse of landmarks.

In the medical application to human aortic models, we have built the Statistical Shape Model out of a training data set of six ascending aorta. The shape instances are preprocessed and fitted in B-spline to perform as the actual input training set instances. The search for a reasonably high-quality SSM is reduced to an correspondence optimization problem, and we propose a novel and efficient scheme for manipulating group-wise shape correspondence, i.e. the direct reparametrization driven by B-spline coefficients. Although our proposed correspondence optimization algorithm currently concentrates on the shape of disk topology, the raw shapes are carefully preprocessed and the interested areas such as the coronary sinuses are kept to form a geometry with cylindrical topology. The generatrix is introduced to transform cylindrical topology shapes to disk topology shapes as valid input into optimization algorithm. Reliable statistical model evaluation measures of generalization ability, specificity and compactness have been studied to show that correspondence is improved and simultaneously an optimized SSM has been achieved. The SSM then can be utilized for a good variety of downstream medical applications such patient-specific modeling and diagnosis where alternatives are either expensive or potentially hazardous.

6.2 Contribution

The method presented in this thesis has the following contributions

- *Direct reparametrization achieved by diffeomorphic deformation through B-splines.* The conventional way of modeling reparametrization function is to concatenate a large number of simple deformation mappings. Due to its undesirable optimization inefficiency, we model the reparametrization function directly by B-spline, and guarantee the diffeomorphism deformation by enforcing bijective constraints, which brings about sizable improvement in optimization efficiency.

- *Full differentiability of objective function (description length) with respect to reparametrization variables.* Our method uses B-spline for both training set shape representation and for reparametrization function modeling, this eliminates the need to use numerical approximation for gradient computation with other alternative geometric forms. Now every step in the statistical modeling pipeline becomes differentiable and a full differentiable objective function gradient becomes possible.

- *Adjoint method for analytical gradient computation.* Under the proposed optimization setting, we derive the highly efficient sensitivity formula by means of adjoint method for the purpose of calculating the full analytical gradient. The derived adjoint sensitivity enjoys superior efficiency compared to numerical counterpart, especially as shape population size increases.

- *Analytical form and efficient numerical quadrature method for computing covariance matrix.* As a critical step in statistical modeling and analysis, the covariance matrix decomposition bears great influence on modes result, and calculating covariance matrix is thus extremely important. Inspired by our proposed use of B-spline forms of shape, we present useful analytical and numerical formulas for its computation, and comprehensively elaborate its properties and influences in the context of statistical shape modeling.

- *Viable application to real medical data for valuable clinical insight.* Our proposed algorithm has been accommodated to two real medical data sets, aorta and proximal femur. The correspondence optimization results suggest noticeable improvement in the quality of the statistical model, and the statistical modes reveal consistent variational pattern as an surgeon or expert would identify.

6.3 Outlook

The proposed method is based on the assumption that boundaries of each shape are already in correspondence, and the algorithm is currently designed for single patch B-spline geometries. In the future, we would explore how to automatically adjust boundary correspondence across shapes, and study how to extend our method to objects of complex topology.

For the experiments on the aorta data, the small pool of six models limits the statistical model's ability to reliably characterize the underlying shape variation pattern of the coronary sinus bump geometries; more aortic models will be collected and incorporated into the training set, which will cover a much wider range of variational pattern. Besides, the extension from simple topology to more complicated topology in 3D case will greatly enhance the method's utility considering the delicate anatomical attachments to aorta such as arteries and valve leaflets etc. Hopefully with these improvements, the SSM will serve as a guidance to understand and disclose the relations among the arteries geometry and the pathological symptoms.

For the application to proximal femur, it's first of all advisable to utilize more femur models for a more faithful shape statistics characterization. More importantly, as the femur impingement involves both the femoral part and the acetabular part, considering the acetabular shape along with the engaging proximal femur would be more reasonable in the diagnosis of impingement.

APPENDIX A

REPRAMETERIZATION OF CURVES BY CAUCHY KERNEL CONCATENATION

Following the formulation in [28], the *Probability Density Function* (PDF) $\rho(u)$ is the sum of n_l Cauchy kernel functions, and the integral of $\rho(u)$ constitutes a *Cumulative Distribution Function* (CDF), which is just the reparameterizatin function.

$$\begin{aligned} \mathcal{R}_S^{\text{Cauchy}}(u) = u' = \int_0^u \rho(u)\, du \\ \rho(u) = & R_{n_l}^{\text{Cauchy}} \circ \cdots \circ R_2^{\text{Cauchy}} \circ R_1^{\text{Cauchy}}(u; c_l, w_l, a_l) \end{aligned} \tag{A.1}$$

where c_l, w_l, a_l are the center, radius and magnitude of the l-th Cauchy kernel. Their sequential superimposition results in the PDF function and leads finally to the reparameterization function CDF. Since the PDF is always positive, the integral of their summation is inherently monotonically increasing, and the diffeomorphic condition is automatically satisfied in this case.

APPENDIX B

REPRAMETERIZATION OF SURFACES BY CLAMED PLATE SPLINE CONCATENATION

A reparameterization function can be represented as a composition of a series of CPS warps; the l-th CPS warp is a localized function defined by CPS center $\mathbf{c}_l$, radius w_l and deformation $\mathbf{a}_l$ [28] as

$$\mathcal{R}_S^{\text{CPS}}(\mathbf{u}) = \mathbf{u}' = R_{n_l}^{\text{CPS}} \circ \cdots \circ R_2^{\text{CPS}} \circ R_1^{\text{CPS}}(\mathbf{u}). \tag{B.1}$$

$$\begin{aligned} R_l^{\text{CPS}}(\mathbf{u}) &= R_l^{\text{CPS}}(\mathbf{u}; \mathbf{c}_l, w_l, \mathbf{a}_l) \\ &= \begin{cases} \mathbf{u} + \mathbf{a}_l \cdot h\left(\dfrac{|\mathbf{u} - \mathbf{c}_l|}{w_l}\right) & \text{if } |\mathbf{u} - \mathbf{c}_l| < w_l, \\ \mathbf{u} & \text{otherwise,} \end{cases} \end{aligned} \tag{B.2}$$

where

$$h(r) = 1 - r^2 + r^2 \ln(r^2). \tag{B.3}$$

The transformation is diffeomorphic with the following constraint

$$|\mathbf{a}| < \frac{1}{|h'(r)|} = w\frac{e}{4}. \tag{B.4}$$

APPENDIX C

JACOBIAN OF A BÉZIER PATCH

Suppose we have a Bézier surface patch $\mathbf{S}(u, v)$ defined by

$$\mathbf{S}(u,v) = \sum_{i=0}^{p} \sum_{j=0}^{q} B_{i,p}(u) B_{j,q}(v) \mathbf{P}_{i,j} \tag{C.1}$$

where p and q are the degree along u- and v-direction, $B_{i,p}(u)$ and $B_{j,q}(v)$ are the Bernstein basis functions and $\mathbf{P}_{i,j}$ is the (i, j)-th control point.

Its Jacobian is then

$$J(u,v) = \det \left[\frac{\partial \mathbf{S}(u,v)}{\partial u}, \frac{\partial \mathbf{S}(u,v)}{\partial v} \right] \tag{C.2}$$

where

$$\begin{aligned} \frac{\partial \mathbf{S}(u,v)}{\partial u} &= \sum_{i=0}^{p-1} \sum_{j=0}^{q} B_{i,p-1}(u) B_{j,q}(v) p \left(\mathbf{P}_{i+1,j} - \mathbf{P}_{i,j} \right), \\ \frac{\partial \mathbf{S}(u,v)}{\partial v} &= \sum_{k=0}^{p} \sum_{l=0}^{q-1} B_{k,p}(u) B_{l,q-1}(v) p \left(\mathbf{P}_{k,l+1} - \mathbf{P}_{k,l} \right) \end{aligned} \tag{C.3}$$

Introducing the notation $\Delta\mathbf{P}_{ij,u} = \mathbf{P}_{i+1,j} - \mathbf{P}_{i,j}$ and $\Delta\mathbf{P}_{kl,v} = \mathbf{P}_{k,l+1} - \mathbf{P}_{k,l}$ gives

$$\begin{aligned} J(u,v) = \sum_{i=0}^{p-1} \sum_{j=0}^{q} \sum_{k=0}^{p} \sum_{l=0}^{q-1} & B_{i,p-1}(u) B_{j,q}(v) B_{k,p}(u) B_{l,q-1}(v) \\ & pq \det \left[\Delta\mathbf{P}_{ij,u}, \Delta\mathbf{P}_{kl,v} \right] \end{aligned} \tag{C.4}$$

After considering the following property of Bernstein polynomial [59]

$$B_{i,p}(u) B_{j,q}(u) = \frac{\binom{p}{i}\binom{q}{j}}{\binom{p+q}{i+j}} B_{i+j,p+q}(u), \tag{C.5}$$

(C.4) becomes

$$\begin{aligned} J(u,v) = & \sum_{i=0}^{p-1} \sum_{j=0}^{q} \sum_{k=0}^{p} \sum_{l=0}^{q-1} \frac{\binom{p-1}{i}\binom{p}{k}}{\binom{2p-1}{i+k}} B_{i+k,2p-1}(u) \\ & \frac{\binom{q}{j}\binom{q-1}{l}}{\binom{2q-1}{j+l}} B_{j+l,2q-1}(v) \; pq \det \left[\Delta\mathbf{P}_{ij,u} \quad \Delta\mathbf{P}_{kl,v} \right] \\ = & \sum_{s=0}^{2p-1} \sum_{t=0}^{2q-1} B_{s,2p-1}(u) B_{t,2q-1}(v) J_{s,t} \end{aligned} \tag{C.6}$$

where

$$J_{s,t} = \sum_{\substack{i+k=s,\\ i\in[0,p-1],\\ k\in[0,p]}} \sum_{\substack{j+l=t,\\ j\in[0,q],\\ l\in[0,q-1]}} \frac{\binom{p-1}{i}\binom{p}{k}}{\binom{2p-1}{i+k}} \frac{\binom{q}{j}\binom{q-1}{l}}{\binom{2q-1}{j+l}} pq \det\left[\Delta\mathbf{P}_{ij,u} \quad \Delta\mathbf{P}_{kl,v}\right] \tag{C.7}$$

BIBLIOGRAPHY

[1] Timothy F Cootes, Christopher J Taylor, David H Cooper, Jim Graham, et al. Active shape models-their training and application. *Computer vision and image understanding*, 61(1):38–59, 1995.

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

[3] Timothy F. Cootes, Gareth J. Edwards, and Christopher J. Taylor. Active appearance models. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 23(6):681–685, 2001.

[4] Nils Hasler, Carsten Stoll, Martin Sunkel, Bodo Rosenhahn, and H-P Seidel. A statistical model of human pose and body shape. In *Computer Graphics Forum*, volume 28, pages 337–346. Wiley Online Library, 2009.

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

[6] Kai-kai Shen, Jurgen Fripp, Fabrice Mériaudeau, Gaël Chételat, Olivier Salvado, and Pierrick Bourgeat. Detecting global and local hippocampal shape changes in alzheimer's disease using statistical shape models. *Neuroimage*, 59(3):2155–2166, 2012.

[7] Stefan Zachow, Hans Lamecker, Barbara Elsholtz, and Michael Stiller. Reconstruction of mandibular dysplasia using a statistical 3d shape model. In *International Congress Series*, volume 1281, pages 1238–1243. Elsevier, 2005.

[8] Gozde Unal, Delphine Nain, Greg Slabaugh, and Tong Fang. Generating shapes by analogies: An application to hearing aid design. *Computer-Aided Design*, 43(1):47–56, 2011.

[9] Rasmus R Jensen, Oline V Olesen, Rasmus R Paulsen, Mike van der Poel, and Rasmus Larsen. Statistical surface recovery: a study on ear canals. In *Mesh Processing in Medical Image Analysis 2012*, pages 49–58. Springer, 2012.

[10] Rémi Blanc, Christof Seiler, Gabor Székely, Lutz-Peter Nolte, and Mauricio Reyes. Statistical model based shape prediction from a combination of direct observations and various surrogates. application to orthopaedic research. *Medical Image Analysis*, 2012.

[11] Rhodri Huw Davies. *Learning shape: optimal models for analysing natural variability.* University of Manchester, 2002.

[12] Oliver Van Kaick, Hao Zhang, Ghassan Hamarneh, and Daniel Cohen-Or. A survey on shape correspondence. In *Computer Graphics Forum*, volume 30, pages 1681–1707. Wiley Online Library, 2011.

[13] Brett Allen, Brian Curless, and Zoran Popović. The space of human body shapes: reconstruction and parameterization from range scans. In *ACM Transactions on Graphics (TOG)*, volume 22, pages 587–594. ACM, 2003.

[14] Natasha Gelfand, Niloy J Mitra, Leonidas J Guibas, and Helmut Pottmann. Robust global registration. In *Proceedings of the third Eurographics symposium on Geometry processing*, pages 197–206, 2005.

[15] Dragomir Anguelov, Praveen Srinivasan, Daphne Koller, Sebastian Thrun, Jim Rodgers, and James Davis. Scape: shape completion and animation of people. In *ACM Transactions on Graphics (TOG)*, volume 24, pages 408–416. ACM, 2005.

[16] Hao Li, Robert W Sumner, and Mark Pauly. Global correspondence optimization for non-rigid registration of depth scans. In *Computer graphics forum*, volume 27, pages 1421–1430. Wiley Online Library, 2008.

[17] Andrew Hill and Christopher J Taylor. Automatic landmark generation for point distribution models. In *Proc. British Machine Vision Conference*, volume 2, pages 429–438, 1994.

[18] AC Kotcheff, Chris J Taylor, et al. Automatic construction of eigenshape models by direct optimization. *Medical Image Analysis*, 2(4):303, 1998.

[19] Rhodri H. Davies, Carole J. Twining, Timothy F. Cootes, John C. Waterton, and Christopher J. Taylor. A minimum description length approach to statistical shape modeling. *Medical Imaging, IEEE Transactions on*, 21(5):525–537, 2002.

[20] Rhodri H Davies, Carole J Twining, P Daniel Allen, Tim F Cootes, and Chris J Taylor. Building optimal 2d statistical shape models. *Image and Vision Computing*, 21(13):1171–1182, 2003.

[21] Rhodri H Davies, Carole J Twining, Timothy F Cootes, and Chris J Taylor. Building 3-d statistical shape models by direct optimization. *Medical Imaging, IEEE Transactions on*, 29(4):961–981, 2010.

[22] Hans Henrik Thodberg. Minimum description length shape and appearance models. In *Information Processing in Medical Imaging*, pages 51–62. Springer, 2003.

[23] Hans Henrik Thodberg and Hildur Olafsdottir. Adding curvature to minimum description length shape models. In *British Machine Vision Conference, BMVC*, 2003.

[24] Anders Ericsson and Kalle Åström. Minimizing the description length using steepest descent. In *Proc. British Machine Vision Conference, Norwich, United Kingdom*, volume 2, pages 93–102, 2003.

[25] Tobias Heimann, Ivo Wolf, Tomos Williams, and Hans-Peter Meinzer. 3d active shape models using gradient descent optimization of description length. In *Information Processing in Medical Imaging*, pages 566–577. Springer, 2005.

[26] Jirı Hladuvka and Katja Bühler. Mdl spline models: Gradient and polynomial reparameterisations. In *17th British Machine Vision Conference*, pages 869–878, 2005.

[27] Timothy F Cootes, Christopher J Taylor, David H Cooper, and Jim Graham. Training models of shape from sets of examples. In *BMVC92*, pages 9–18. Springer, 1992.

[28] Rhodri Davies, Carole Twining, and Chris J Taylor. *Statistical models of shape: Optimisation and evaluation.* Springerverlag London, 2008.

[29] Colin Goodall. Procrustes methods in the statistical analysis of shape. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 285–339, 1991.

[30] Rhodri H Davies, Carole J Twining, Chris Taylor, et al. Groupwise surface correspondence by optimization: representation and regularization. *Medical image analysis*, 12(6):787, 2008.

[31] Ian Jolliffe. *Principal component analysis*. Wiley Online Library, 2005.

[32] Les A Piegl and Wayne Tiller. *The NURBS book*. Springer Verlag, 1997.

[33] Michael S Floater. Parametrization and smooth approximation of surface triangulations. *Computer aided geometric design*, 14(3):231–250, 1997.

[34] Gerald E Farin. *Curves and surfaces for CAGD: a practical guide*. Morgan Kaufmann, 2002.

[35] Xiaoping Qian and Ole Sigmund. Isogeometric shape optimization of photonic crystals via coons patches. *Computer Methods in Applied Mechanics and Engineering*, 200(25):2237–2255, 2011.

[36] Xilu Wang and Xiaoping Qian. An optimization approach for constructing trivariate b-spline solids. *Computer-Aided Design*, 46:179–191, 2014.

[37] Knut Mørken. Some identities for products and degree raising of splines. *Constructive Approximation*, 7(1):195–208, 1991.

[38] Hongwei Lin, Kai Tang, Ajay Joneja, and Hujun Bao. Generating strictly non-self-overlapping structured quadrilateral grids. *Computer-Aided Design*, 39(9):709–718, 2007.

[39] Thomas W Sederberg and Scott R Parry. Free-form deformation of solid geometric models. In *ACM Siggraph Computer Graphics*, volume 20, pages 151–160. ACM, 1986.

[40] Daniel Rueckert, Luke I Sonoda, Carmel Hayes, Derek LG Hill, Martin O Leach, and David J Hawkes. Nonrigid registration using free-form deformations: application to breast mr images. *Medical Imaging, IEEE Transactions on*, 18(8):712–721, 1999.

[41] Torsten Rohlfing, Calvin R Maurer Jr, David A Bluemke, and Michael A Jacobs. Volume-preserving nonrigid registration of mr breast images using free-form deformation with an incompressibility constraint. *Medical Imaging, IEEE Transactions on*, 22(6):730–741, 2003.

[42] Michaël Sdika. A fast nonrigid image registration with constraints on the jacobian using large scale constrained optimization. *Medical Imaging, IEEE Transactions on*, 27(2):271–281, 2008.

[43] Marc Modat, Gerard R Ridgway, Zeike A Taylor, Manja Lehmann, Josephine Barnes, David J Hawkes, Nick C Fox, and Sébastien Ourselin. Fast free-form deformation using graphics processing units. *Computer methods and programs in biomedicine*, 98(3):278–284, 2010.

[44] Vincent Camion and Laurent Younes. Geodesic interpolating splines. In *Energy Minimization Methods in Computer Vision and Pattern Recognition*, pages 513–527. Springer, 2001.

[45] Laurent Younes. *Shapes and diffeomorphisms*, volume 171. Springer, 2010.

[46] Eitan Sharon and David Mumford. 2d-shape analysis using conformal mapping. *International Journal of Computer Vision*, 70(1):55–75, 2006.

[47] Colin J Cotter. The variational particle-mesh method for matching curves. *Journal of Physics A: Mathematical and Theoretical*, 41(34):344003, 2008.

[48] Colin J Cotter, Allan Clark, and Joaquim Peiró. A reparameterisation based approach to geodesic constrained solvers for curve matching. *International journal of computer vision*, 99(1):103–121, 2012.

[49] Martin Bauer, Martins Bruveris, Stephen Marsland, and Peter W Michor. Constructing reparametrization invariant metrics on spaces of plane curves. *arXiv preprint arXiv:1207.5965*, 2012.

[50] Niles A Pierce and Michael B Giles. Adjoint recovery of superconvergent functionals from pde approximations. *SIAM review*, 42(2):247–264, 2000.

[51] Michael B Giles and Endre Süli. Adjoint methods for pdes: a posteriori error analysis and postprocessing by duality. *Acta Numerica*, 11:145–236, 2002.

[52] Michael B Giles and Niles A Pierce. An introduction to the adjoint approach to design. *Flow, turbulence and combustion*, 65(3-4):393–415, 2000.

[53] JRRA Martins and Nicholas MK Poon. On structural optimization using constraint aggregation. In *6th world congress on structural and multidisciplinary optimization. ISSMO, Rio de Janeiro, Brazil*, 2005.

[54] J París, F Navarrina, I Colominas, and M Casteleiro. Block aggregation of stress constraints in topology optimization of structures. *Advances in Engineering Software*, 41(3):433–441, 2010.

[55] RJ Yang and CJ Chen. Stress-based topology optimization. *Structural Optimization*, 12(2-3):98–105, 1996.

[56] M. B. Stegmann and D. D. Gomez. A brief introduction to statistical shape analysis, mar 2002. Images, annotations and data reports are placed in the enclosed zip-file.

[57] William T Tutte. How to draw a graph. *Proc. London Math. Soc*, 13(3):743–768, 1963.

[58] Shin Yoshizawa, Alexander Belyaev, and Hans-Peter Seidel. A fast and simple stretch-minimizing mesh parameterization. In *Shape Modeling Applications, 2004. Proceedings*, pages 200–208. IEEE, 2004.

[59] Rida T Farouki and VT Rajan. Algorithms for polynomials in bernstein form. *Computer Aided Geometric Design*, 5(1):1–26, 1988.

[60] Eugene TY Lee. Choosing nodes in parametric curve interpolation. *Computer-Aided Design*, 21(6):363–370, 1989.

[61] Michael S Sacks, Ali Mirnajafi, Wei Sun, and Paul Schmidt. Bioprosthetic heart valve heterograft biomaterials: structure, mechanical behavior and computational simulation. *Expert review of medical devices*, 3(6):817–834, 2006.

[62] Ben H Thacker, David S Riha, Simeon HK Fitch, Luc J Huyse, and Jason B Pleming. Probabilistic engineering analysis using the nessus software. *Structural Safety*, 28(1):83–107, 2006.

[63] Mark A Baldwin, Peter J Laz, Joshua Q Stowe, and Paul J Rullkoetter. Efficient probabilistic representation of tibiofemoral soft tissue constraint. *Computer Methods in Biomechanics and Biomedical Engineering*, 12(6):651–659, 2009.

[64] Peter J Laz, Saikat Pal, Aaron Fields, Anthony J Petrella, and Paul J Rullkoetter. Effects of knee simulator loading and alignment variability on predicted implant mechanics: a probabilistic study. *Journal of orthopaedic research*, 24(12):2212–2221, 2006.

[65] Peter J Laz, Saikat Pal, Jason P Halloran, Anthony J Petrella, and Paul J Rullkoetter. Probabilistic finite element prediction of knee wear simulator mechanics. *Journal of biomechanics*, 39(12):2303–2310, 2006.

[66] Saikat Pal, Joseph E Langenderfer, Joshua Q Stowe, Peter J Laz, Anthony J Petrella, and Paul J Rullkoetter. Probabilistic modeling of knee muscle moment arms: effects of methods, origin–insertion, and kinematic variability. *Annals of biomedical engineering*, 35(9):1632–1642, 2007.

[67] IL Dryden and KV Mardia. *Statistical analysis of shape*. Wiley, 1998.

[68] Timothy F Cootes, Andrew Hill, Christopher J Taylor, and Jane Haslam. Use of active shape models for locating structures in medical images. *Image and vision computing*, 12(6):355–365, 1994.

[69] Markus Fleute and Stéphane Lavallée. Building a complete surface model from sparse data using statistical shape models: Application to computer assisted knee surgery. In *Medical Image Computing and Computer-Assisted InterventationMICCAI98*, pages 879–887. Springer, 1998.

[70] Alejandro F Frangi, Daniel Rueckert, Julia A Schnabel, and Wiro J Niessen. Automatic construction of multiple-object three-dimensional statistical shape models: Application to cardiac modeling. *Medical Imaging, IEEE Transactions on*, 21(9):1151–1166, 2002.

[71] AP King, JM Blackall, GP Penney, and DJ Hawkes. Tracking liver motion using 3-d ultrasound and a surface based statistical shape model. In *Mathematical Methods in Biomedical Image Analysis, 2001. MMBIA 2001. IEEE Workshop on*, pages 145–152. IEEE, 2001.

[72] Fani Deligianni, Adrian James Chung, and Guang-Zhong 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):1462–1471, 2006.

[73] Yonghong Shi, Feihu Qi, Zhong Xue, Liya Chen, Kyoko Ito, Hidenori Matsuo, and Dinggang Shen. Segmenting lung fields in serial chest radiographs using both population-based and patient-specific shape statistics. *Medical Imaging, IEEE Transactions on*, 27(4):481–494, 2008.

[74] P Danny Allen, Jim Graham, Damian JJ Farnell, Elizabeth J Harrison, Reinhilde Jacobs, Kety Nicopolou-Karayianni, Christina Lindh, Paul F van der Stelt, Keith Horner, and Hugh Devlin. Detecting reduced bone mineral density from

dental radiographs using statistical shape models. *Information Technology in Biomedicine, IEEE Transactions on*, 11(6):601–610, 2007.

[75] Paul P Smyth, Christopher J Taylor, and Judith E Adams. Vertebral shape: Automatic measurement with active shape models1. *Radiology*, 211(2):571–578, 1999.

[76] Hugh Devlin, Philip D Allen, Jim Graham, Reinhilde Jacobs, Kety Karayianni, Christina Lindh, Paul F van der Stelt, Elizabeth Harrison, JE Adams, Susan Pavitt, et al. Automated osteoporosis risk assessment by dentists: a new pathway to diagnosis. *Bone*, 40(4):835–842, 2007.

[77] Lucas Eduardo Ritacco, L Aponte-Tinao, DL Muscolo, FG de Quirós, and I Nozomu. Three-dimensional morphometric analysis of the distal femur: a validity method for allograft selection using a virtual bone bank. *Studies in health technology and informatics*, 160(Pt 2):1287–1290, 2009.

[78] Michael D Harris, Manasi Datar, Ross T Whitaker, Elizabeth R Jurrus, Christopher L Peters, and Andrew E Anderson. Statistical shape modeling of cam femoroacetabular impingement. *Journal of Orthopaedic Research*, 31(10):1620–1626, 2013.

[79] Peter D Grünwald. *The minimum description length principle.* The MIT Press, 2007.

[80] Alla Sheffer, Emil Praun, and Kenneth Rose. Mesh parameterization methods and their applications. *Foundations and Trends® in Computer Graphics and Vision*, 2(2):105–171, 2006.

[81] Joseph o'Rourke. *Computational geometry in C.* Cambridge university press, 1998.

[82] Paul J Besl and Neil D McKay. Method for registration of 3-d shapes. In *Robotics-DL tentative*, pages 586–606. International Society for Optics and Photonics, 1992.

[83] Shen Dong, Scott Kircher, and Michael Garland. Harmonic functions for quadrilateral remeshing of arbitrary manifolds. *Computer aided geometric design*, 22(5):392–423, 2005.

[84] Tobias Martin, Elaine Cohen, and Robert M Kirby. Volumetric parameterization and trivariate b-spline fitting using harmonic functions. *Computer Aided Geometric Design*, 26(6):648–664, 2009.