

Direct Diffeomorphic Reparameterization for Correspondence Optimization in Statistical Shape Modeling

Kang Li

Mechanical, Materials and Aerospace Engineering Department
Illinois Institute of Technology
Chicago, IL 60616

Xiaoping Qian*

Department of Mechanical Engineering
The University of Wisconsin-Madison
Madison, WI 53706

March 2, 2015

Abstract

In this paper, we propose 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 reparameterization 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 the 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 sensitivity 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.

Keywords Statistical Shape Model, Shape Correspondence, Direct Reparameterization , Adjoint Method

1 Introduction

A *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].

*Email addresses: kli@hawk.iit.edu (K. Li) and qian@engr.wisc.edu (X. Qian: corresponding author).

It has since seen many other applications including facial recognition [3], computer animation [4], medical diagnosis [5, 6], patient-specific modeling [7–10] and human body modeling [11], to name but a few. Finding correspondence across all shape instances is a fundamental task in building 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 [12]. 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 manipulating correspondence either in the object space or in the parameter space. Thus far, one of the most common approaches to achieving shape correspondence is through deforming in the object space a template shape to 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. Group-wise registration has also been developed [13]. In such a deformation based correspondence manipulation approach, the deformed landmarks may not actually lie on the object shape before the optimization convergence is achieved. Further, the deformation 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 a 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 [14–17]. However, these geometric descriptors and the regularization constraints are not necessarily a good basis for correspondence.

An alternative is to manipulate the correspondence through reparameterizing the shapes in the parameter space. For example, in [18], objects of spherical topology are mapped to a sphere and correspondence is manipulated through concatenations of symmetric theta transformations on the spherical map. Reparameterization of shapes in the parameter space thus allows convenient manipulation of correspondence of surface points by simply changing point parameters. Reparameterized points always lie on the object shape during the optimization process. As such, the reparameterization lends itself to a more principled approach for establishing correspondence: optimizing the quality of the resulting statistical models. During the past few years, SSM quality measures have evolved from the model covariance trace [19], to the model covariance determinant [20], and finally to the *Description Length*(DL) [18, 21, 22] and its simplification [23] or variants [24]. This information theoretic objective function of description length has shown to be an effective measure [21] 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 [21], 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 [25, 26]. 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 [27], 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 [28]. In our approach, each landmark point $\mathbf{S}(\mathbf{u})$ in a given shape is changed to $\mathbf{S}(\mathbf{D}(\mathbf{u}))$ 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 $\mathbf{S}(\mathbf{u})$ 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 illustrates the proposed idea. A group of hand contours are shown in Figure 1(a). Each shape is represented with B-splines, and Figure 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(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(d) where each red point represents a B-spline coefficient for the reparameterization. The landmarks are redistributed as shown in Figure 1(e) after the reparameterization. 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.

The salient feature of this approach is as follows

- *Diffeomorphic reparameterization through B-splines.* Existing technique [18] for reparameterization concatenates a series of simple homeomorphic mappings. One optimization run with this reparameterization technique leads to the deformation of a local parametric region and it cannot provide any information on the search direction for subsequent local deformations in other regions. Therefore it requires the concatenation of a large number of simple mappings and causes severe inefficiency (See Section 5.1). Instead of concatenation of many local mappings, we propose the use of single B-spline function to directly represent the diffeomorphic reparameterization $\mathbf{D}(\mathbf{u})$ for the parameterization \mathbf{u} of each shape instance $\mathbf{S}(\mathbf{u})$. The 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 reparameterization) 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

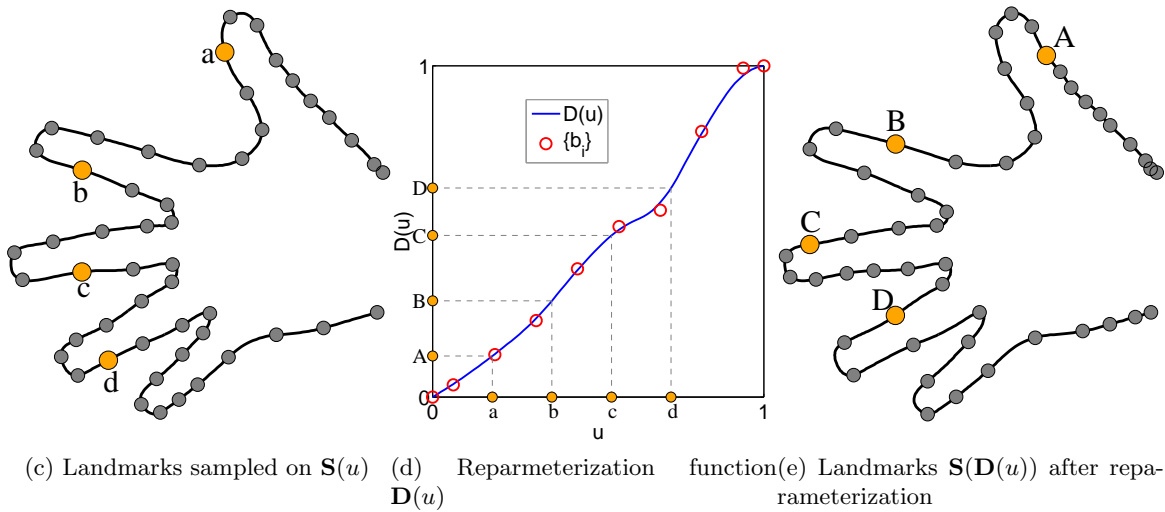
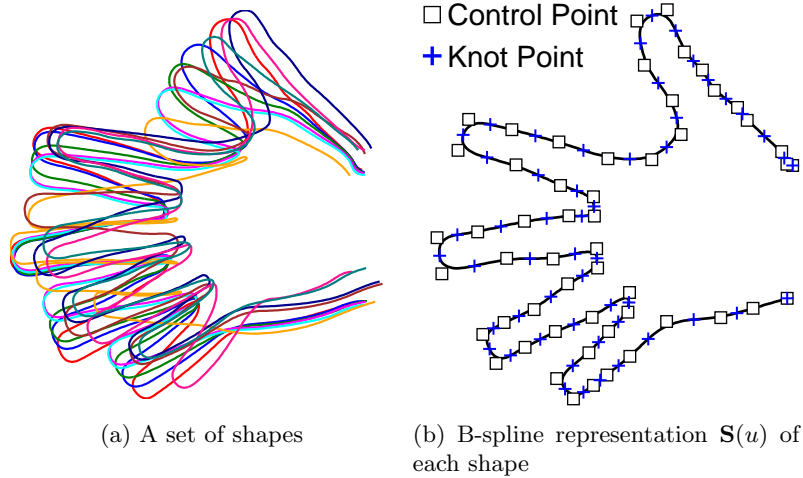


Figure 1: Landmark manipulation by reparameterization for improving shape correspondence in statistical shape modeling.

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 \mathbf{b} , eigen values 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.

The remainder of this paper is organized as follows. Section 2 reviews the basic computing procedures in statistical shape modeling and the role of diffeomorphic reparameterization in SSM. Section 3 details our proposed approach on direct diffeomorphic reparameterization for shape correspondence. Section 4 presents the developed optimization techniques with experimental results in Section 5. This paper is concluded in Section 6.

2 Background on statistical shape modeling

In this section, we review the basic computing procedures in statistical shape modeling and the role of diffeomorphic reparameterization in SSM.

Statistical Shape Model was initially called *Point Distribution Model* (PDM) [28], and it requires each shape instance in the training set be represented by a set of points, known as the *landmarks* [19]. 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 [29]. The statistical modeling framework requires that the same number of landmarks sampled on all shapes across the training set.

Suppose a training set $\{\mathbb{T}_i\}$ ($i = 1, 2, \dots, n_S$) comprises n_S shape instances and each is represented by n_P landmarks. The j -th landmark of the i -th instance is $\mathbf{x}_i^{(j)} = [x_i^{(j)}, y_i^{(j)}, z_i^{(j)}]^T \in \mathbb{R}^3$. Owing to the correspondence assumption of landmarks, all the n_S landmarks $\{\mathbf{x}_i^{(j)}\}$ ($i = 1, 2, \dots, n_S$) 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:

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

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

$$\mathbf{X}^S \doteq [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_{n_S}]. \quad (1)$$

In order to align a group of shapes stored in \mathbf{X}^S , the *Generalized Procrustes Analysis* (GPA) [30] operation is conducted, denoted by an alignment operator as below

$$\mathbf{X}^A = \mathcal{A}(\mathbf{X}^S). \quad (2)$$

More 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

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

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 [31].

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

$$\mathbf{E}_{\mu\nu} = \frac{1}{(n_S - 1)A} \int [\mathbf{S}_\mu(\mathbf{u}) - \bar{\mathbf{S}}(\mathbf{u})] \cdot [\mathbf{S}_\nu(\mathbf{u}) - \bar{\mathbf{S}}(\mathbf{u})] dA(\mathbf{u}), \quad (3)$$

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 defined as follows

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

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

$$\mathbf{E}_{\mu\nu} = \frac{1}{(n_S - 1)n_P} \sum_{i=1}^{n_S} (\mathbf{x}_i - \bar{\mathbf{x}})_\mu (\mathbf{x}_i - \bar{\mathbf{x}})_\nu. \quad (4)$$

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 [21]

$$\mathbf{E} = \frac{1}{(n_S - 1)n_P} \mathbf{X}_c^T \mathbf{X}_c, \quad (5)$$

where \mathbf{X}_c is defined by

$$\mathbf{X}_c \doteq [\mathbf{x}_1^A - \bar{\mathbf{x}}, \mathbf{x}_2^A - \bar{\mathbf{x}}, \dots, \mathbf{x}_{n_S}^A - \bar{\mathbf{x}}], \quad (6)$$

and \mathbf{x}_i^A is the i -th shape vector after alignment operation \mathcal{A} , i.e. the component of \mathbf{X}^A in (2); the *mean shape vector* of SSM is

$$\bar{\mathbf{x}} = \frac{1}{n_S} \sum_{i=1}^{n_S} \mathbf{x}_i^A. \quad (7)$$

The *Principal Component Analysis* (PCA) [32] is then conducted to extract the principal modes of shape variability via the eigenvalue decomposition of the covariance matrix

$$\mathbf{E}\mathbf{v}_m = \lambda_m \mathbf{v}_m \quad (m = 1, 2, \dots, n_S - 1), \quad (8)$$

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 \tilde{m} ($\tilde{m} \leq n_S - 1$) modes

$$\mathbf{x} \approx \bar{\mathbf{x}} + \sum_{m=1}^{\tilde{m}} \beta_m \mathbf{v}_m, \quad (9)$$

where the m -th mode parameter is found by projection

$$\beta_m = (\mathbf{x} - \bar{\mathbf{x}})^T \mathbf{v}_m. \quad (10)$$

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 (5) and (8) is compactly written as

$$\boldsymbol{\lambda} = \mathcal{C}(\mathbf{X}^A). \quad (11)$$

The objective function of Description Length was originally derived by Davies in [21] and elaborated in [29]. A simplified version presented in [23] defined as below is used in our correspondence optimization:

$$f \doteq \sum_{m=1}^{n_S-1} L_m, \quad (12)$$

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

$$\lambda_{\text{cut}} = \frac{2 l_{\min}}{r_{\max}}, \quad (13)$$

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.

3 Direct diffeomorphic reparameterization for correspondence optimization

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 paper, we choose to use B-splines to represent the shapes. A B-spline curve of degree d is defined by

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

where $B_{k,d}$ is the B-spline basis function [33] of degree d associated with the k -th control points \mathbf{P}_k recursively defined on a non-decreasing knot vector $\Xi = \{\bar{\xi}_0, \bar{\xi}_1, \dots, \bar{\xi}_{n+d+1}\}$ as

$$\begin{aligned} B_{k,d}(u) &= \frac{(u - \bar{\xi}_k) B_{k,d-1}(u)}{\bar{\xi}_{k+d} - \bar{\xi}_k} + \frac{(\bar{\xi}_{k+d+1} - u) B_{k+1,d-1}(u)}{\bar{\xi}_{k+d+1} - \bar{\xi}_{k+1}}, \\ B_{k,0}(u) &= \begin{cases} 1 & \bar{\xi}_k \leq u \leq \bar{\xi}_{k+1}, \\ 0 & \text{otherwise.} \end{cases} \end{aligned} \quad (14)$$

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

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

where $B_{k,d}$ and $B_{l,e}$ following the basis definition in (14) is associated with the (k, l) -th control point $\mathbf{P}_{k,l}$, and the knot vectors along the two parametric directions are $\Xi = \{\bar{\xi}_0, \bar{\xi}_1, \dots, \bar{\xi}_{n_u+d+1}\}$ and $\mathbf{H} = \{\bar{\eta}_0, \bar{\eta}_1, \dots, \bar{\eta}_{n_v+e+1}\}$.

3.1 B-spline based direct reparameterization

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

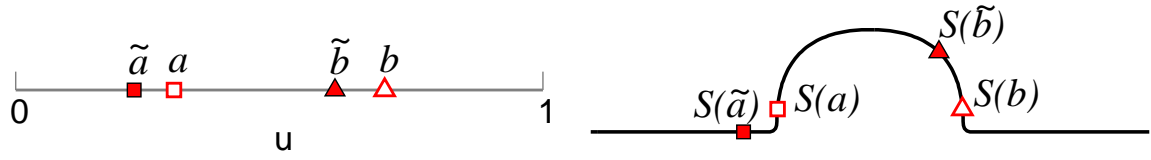
3.1.1 Reparameterization of parametric curves

Fig. 2 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 concatenation of simple mappings. For example, the reparameterization function $D(u)$ shown in Fig. 2 is represented in Fig. 3(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 [29]; See Appendix I for details regarding Cauchy kernel and its concatenation. .

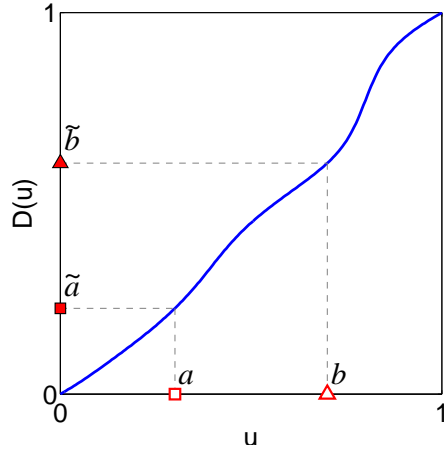
In this paper, we propose to directly represent $D(u)$ as a single B-spline function as shown in Fig. 3(b) where empty circles represent B-spline coefficients $\{b_i\}$. The reparameterization function $D(u)$ for a parametric curve can be represented by a B-spline with n_b coefficients

$$D(u) = \sum_{i=0}^{n_b-1} B_{i,p}(u) b_i, \quad 0 \leq u \leq 1; \quad (15)$$

where $B_{i,p}$ is the B-spline basis function [33] of degree p associated with the i -th B-spline coefficient b_i , and it is recursively defined on a non-decreasing knot vector $\mathbf{U} = \{\bar{u}_0, \bar{u}_1, \dots, \bar{u}_{n+p+1}\}$.

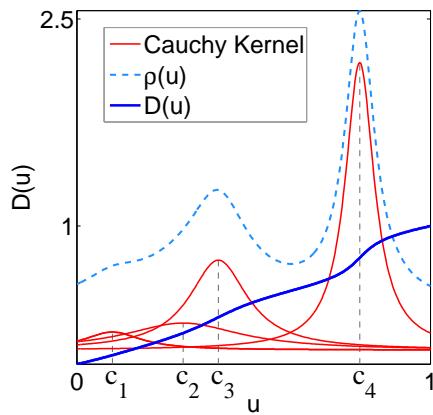


(a) Two landmark points $u_a = 0.30, u_b = 0.70$ in parameter domain repositioned to $\tilde{u}_a = 0.22, \tilde{u}_b = 0.60$ (b) Associated landmarks $S(0.30), S(0.70)$ redistributed to $S(0.22), S(0.60)$ in physical domain

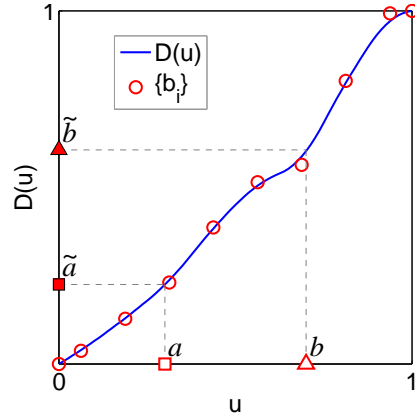


(c) Raparameterization function

Figure 2: 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 show in circles

Figure 3: Representing reparameterization function for 2D curves: Concatenation of functions vs. single function

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 $D(1) = 1$.

3.1.2 Reparameterization of parametric surfaces

A reparameterization of a 3D surface is illustrated in Fig. 4, where cubes and spheres 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 Fig. 4(c), and its parametric grid is shown Fig. 4(d).

Such a reparameterization field can be represented by the concatenation of simple mappings. For example, two Clamped Plate Spline (CPS) warps [29] are applied sequentially in the parameter domain shown in Fig. 5(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.

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 Fig. 5(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

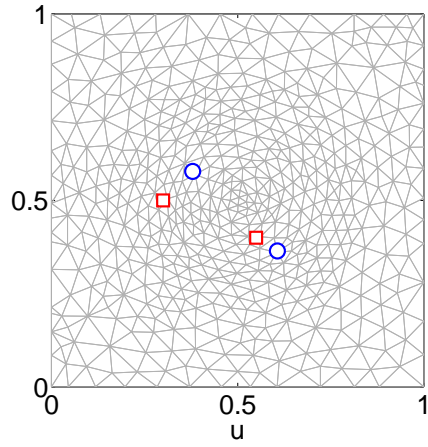
$$\begin{aligned} D^u(u, v) &= \sum_{i=0}^{n_{b1}-1} \sum_{j=0}^{n_{b2}-1} B_{i,p}(u) B_{j,q}(v) b_{i,j}^u, \\ D^v(u, v) &= \sum_{i=0}^{n_{b1}-1} \sum_{j=0}^{n_{b2}-1} B_{i,p}(u) B_{j,q}(v) b_{i,j}^v, \\ 0 &\leq u, v \leq 1; \end{aligned} \tag{16}$$

where $B_{i,p}$ and $B_{j,q}$ are the B-spline basis functions (14) of degree p and q associated with the (i, j) -th B-spline coefficient 2-tuple $\mathbf{b}_{i,j} = (b_{i,j}^u, b_{i,j}^v)$; 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 .

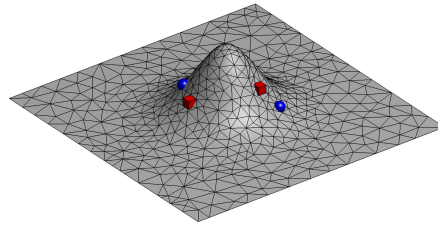
The four boundaries of all the shapes are assumed to be in correspondence already. With this assumption, the four boundaries of the square parametric domain are fixed during reparameterization, i.e.

$$\begin{aligned} D^u(0, v) &= 0, & D^u(1, v) &= 1, \\ D^v(u, 0) &= 0, & D^v(u, 1) &= 1. \end{aligned} \tag{17}$$

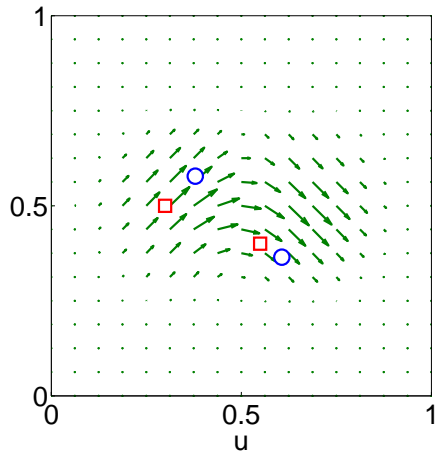
Therefore, the two knot vectors are chosen to be of clamped type and the B-spline coefficients at the four boundaries are either 0 and 1.



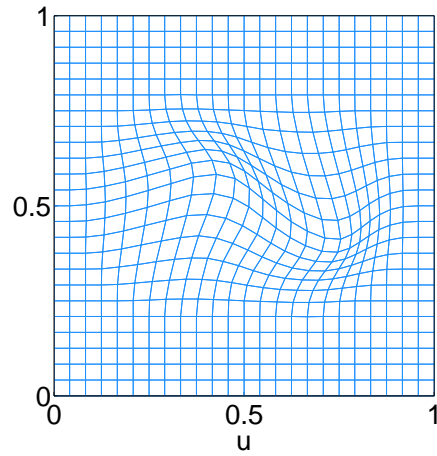
(a) Sampled parametric points



(b) Sampled surface points

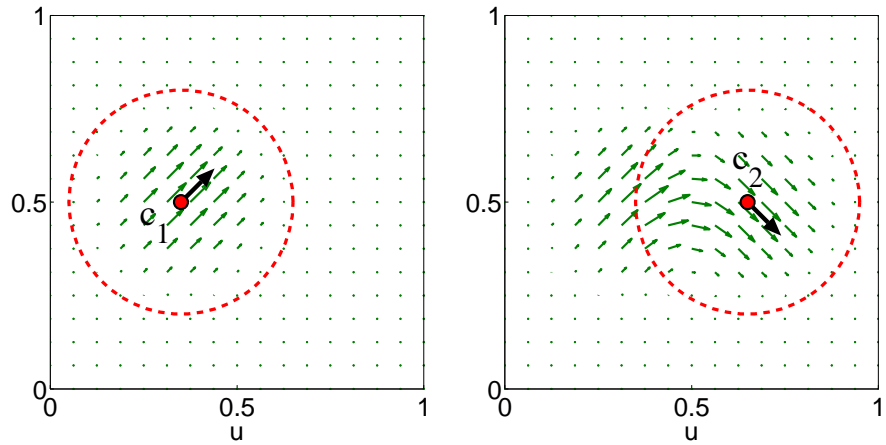


(c) Reparameterization function $\mathbf{D}(\mathbf{u}) = (D^u, D^v)$

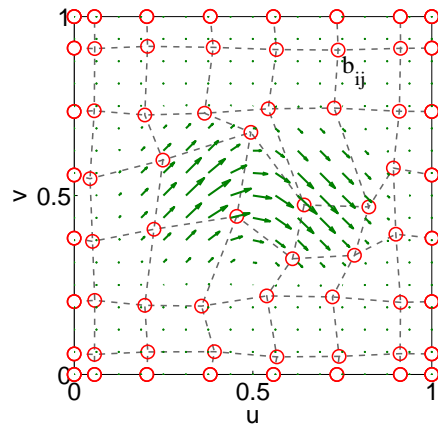


(d) Reparameterized parametric grid

Figure 4: Reparameterization vector field $\mathbf{D}(\mathbf{u})$ and sample redistribution on a 3D surface: cubes and spheres respectively represent points before and after reparameterization.



(a) Concatenation of 2 sequential CPSs: step 1 at c_1 (b) Concatenation of 2 sequential CPSs: step 2 at c_2



(c) Reparameterization by B-splines

Figure 5: Representing reparameterization vector field: concatenation of simple mappings vs. single B-spline based mapping.

3.2 Diffeomorphism of reparameterization

3.2.1 Curve case

Setting $dD(u)/du > 0$ in (15) 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

$$b_i - b_{i+1} < 0, \quad i = 0, 1, \dots, n_b - 2. \quad (18)$$

3.2.2 Surface case

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

$$J(\mathbf{D}(\mathbf{u})) = \begin{vmatrix} \frac{\partial D^u(u,v)}{\partial u} & \frac{\partial D^u(u,v)}{\partial v} \\ \frac{\partial D^v(u,v)}{\partial u} & \frac{\partial D^v(u,v)}{\partial v} \end{vmatrix} > 0, \quad \forall (u, v) \in [0, 1]. \quad (19)$$

Assuming there are $n_{b1} \times n_{b2}$ B-spline coefficient tuples $\mathbf{b}_{i,j}$ for representing the reparameterization in (16). Due to the boundary constraint (17), 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 (19) 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 $J(\mathbf{u})$ (19) is positive so that the reparameterization $\mathbf{D}(\mathbf{u})$ is diffeomorphic. The scalar Jacobian field $J(\mathbf{u})$ defined in (19) for the B-spline based reparameterization (16) 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

$$\begin{aligned} J(u, v) &= \det \begin{bmatrix} \frac{\partial \mathbf{D}(u, v)}{\partial u} & \frac{\partial \mathbf{D}(u, v)}{\partial v} \end{bmatrix}, \\ \frac{\partial \mathbf{D}(u, v)}{\partial u} &= \sum_{i=0}^{n_{b1}-2} \sum_{j=0}^{n_{b2}-1} B_{i,p-1}(u) B_{j,q}(v) \gamma_i (\mathbf{b}_{i+1,j} - \mathbf{b}_{i,j}), \\ \frac{\partial \mathbf{D}(u, v)}{\partial v} &= \sum_{k=0}^{n_{b1}-1} \sum_{l=0}^{n_{b2}-2} B_{k,p}(u) B_{l,q-1}(v) \eta_l (\mathbf{b}_{k,l+1} - \mathbf{b}_{k,l}), \end{aligned} \quad (20)$$

where: $\gamma_i = \frac{p}{\bar{u}_{p+i+1} - \bar{u}_{i+1}}$; $\eta_l = \frac{q}{v_{q+l+1} - v_{l+1}}$. Using the notation $\Delta \mathbf{b}_{i,j}^u = (\mathbf{b}_{i+1,j} - \mathbf{b}_{i,j})$ and $\Delta \mathbf{b}_{k,l}^v = (\mathbf{b}_{k,l+1} - \mathbf{b}_{k,l})$ 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

$$\begin{aligned}
J(\mathbf{u}) &= J(u, v) = \sum_{i=0}^{n_{b1}-2} \sum_{j=0}^{n_{b1}-1} \sum_{k=0}^{n_{b1}-1} \sum_{l=0}^{n_{b2}-2} B_{i,p-1}(u) B_{j,q}(v) \\
&\quad B_{k,p}(u) B_{l,q-1}(v) \gamma_s \eta_l \det [\Delta \mathbf{b}_{i,j}^u \quad \Delta \mathbf{b}_{k,l}^v] \\
&= \sum_{s=0}^{2n_{b1}-3} \sum_{t=0}^{2n_{b2}-3} B_{s,2p-1}(u) B_{t,2q-1}(v) J_{s,t}^{BSP}(\{\mathbf{b}\}),
\end{aligned} \tag{21}$$

where $J_{s,t}^{BSP}$ is the B-spline coefficient of the B-spline form of the Jacobian $J(\mathbf{u})$ for the reparameterization $\mathbf{D}(\mathbf{u})$ in (16); each $J_{s,t}^{BSP}$ 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 (21) is positive, i.e.

$$\begin{aligned}
&J_{s,t}^{BSP}(\{\mathbf{b}\}) > 0, \\
&s = 0, 1, \dots, n_{J1} - 1; \quad t = 0, 1, \dots, n_{J2} - 1,
\end{aligned} \tag{22}$$

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 coefficients $\{\tilde{\mathbf{b}}\}$) 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 $\{J_{s,t}^{BEZ}\}$

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 coefficients $\{\tilde{\mathbf{b}}\}$ 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

$$\begin{aligned}
&J_{s,t}^A = J_{s,t}^{BEZ}(\{\tilde{\mathbf{b}}\}) > 0 \\
&s = 1, \dots, n_{J1}; \quad t = 1, \dots, n_{J2}
\end{aligned} \tag{23}$$

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 (20). The parameter points chosen are usually those associated with the parametric locations of the landmarks. Suppose there are $n_{J_1} = n_{P_1}$ and $n_{J_2} = n_{P_2}$ 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

$$J_{s,t}^B = J(\mathbf{u}_{s,t}, \{\mathbf{b}\}) = \det \left[\frac{\partial \mathbf{D}(\mathbf{u}_{s,t})}{\partial u} \quad \frac{\partial \mathbf{D}(\mathbf{u}_{s,t})}{\partial v} \right] > 0 \quad (24)$$

$$s = 1, \dots, n_{J_1}; t = 1, \dots, n_{J_2},$$

where n_{J_1} and n_{J_2} 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 $n_J = n_{J_1}n_{J_2} = n_{P_1}n_{P_2}$ 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 paper 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 5.

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 in for example Fig 2b are akin to fingerprints observed in [46]. There are also alternative for diffeomorphic reparameterization for curves as reported in [47] [48] [49].

4 Optimization techniques

4.1 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:

$$\min_{\mathbf{b}} f(\mathbf{b}) = \sum_{\lambda_i \geq \lambda_{\text{cut}}} \left[1 + \log \frac{\lambda_k(\mathbf{b})}{\lambda_{\text{cut}}} \right] + \sum_{\lambda_k < \lambda_{\text{cut}}} \frac{\lambda_k(\mathbf{b})}{\lambda_{\text{cut}}} \quad (25a)$$

$$\text{s.t.} \quad [\mathbf{C}^T(\mathbf{b})\mathbf{C}(\mathbf{b})] \mathbf{v}_k(\mathbf{b}) = \lambda_k(\mathbf{b})\mathbf{v}_k(\mathbf{b}) \quad (25b)$$

$$\mathbf{v}_k^T(\mathbf{b})\mathbf{v}_k(\mathbf{b}) = 1, \quad k = 1, \dots, n_S \quad (25c)$$

$$g_l(\mathbf{b}) < 0, \quad l = 1, \dots, n_G \quad (25d)$$

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 (25b) and (25c). 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 (25d) represents the diffeomorphic conditions, i.e. (18) for curves and (23) (24) for surfaces, each of which is a function of optimization variables \mathbf{b} .

The optimization formulation given in (25) 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 (23) (24), which will be detailed in Section 4.3.1.

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 (25a) 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

$$\frac{df}{db_r} = \sum_i \frac{\partial f}{\partial \lambda_i} \sum_j \frac{\partial \lambda_i}{\partial x_j^A} \sum_k \frac{\partial x_j^A}{\partial x_k^S} \sum_l \frac{\partial x_k^S}{\partial \mathbf{D}(\mathbf{u}_l)} \sum_r \frac{\partial \mathbf{D}(\mathbf{u}_l)}{\partial b_r} \quad (26)$$

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 (15) and (16) 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 paper, 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.

4.2 Analytical gradient of objective function

The total sensitivity from (25a) is just

$$\frac{df(\mathbf{b})}{db_r} = \sum_i \frac{\partial f[\mathbf{b}, \boldsymbol{\lambda}(\mathbf{b})]}{\partial \lambda_i} \frac{\partial \lambda_i(\mathbf{b})}{\partial b_r} \quad (27)$$

where $r = 1, \dots, n_D$; 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 (25), we can obtain the objective sensitivity in (27) 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.

4.2.1 Direct sensitivity

The direct sensitivity computes the gradient of the cost function by directly differentiating the cost function (25a). That is,

$$\frac{\partial f(\mathbf{b})}{\partial b_r} = \sum_{m=1}^{n_S} \frac{\partial f}{\partial \lambda_m} \frac{\partial \lambda_m}{\partial b_r}. \quad (28)$$

The term $\partial \lambda_m / \partial b_r$ ($m = 1, \dots, n_S$) is obtained by differentiating equations (25b) and (25c) with respect b_r , which leads to the following resulting linear system

$$\begin{bmatrix} 0 & \mathbf{v}_m^T \\ \mathbf{v}_m & \lambda_m \mathbf{I}_{n_S} - \mathbf{C}^T \mathbf{C} \end{bmatrix} \begin{bmatrix} \frac{\partial \lambda_m}{\partial b_r} \\ \frac{\partial \mathbf{v}_m}{\partial b_r} \end{bmatrix} = \begin{bmatrix} 0 \\ \frac{\partial (\mathbf{C}^T \mathbf{C})}{\partial b_r} \mathbf{v}_m \end{bmatrix}. \quad (29)$$

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 (28) yields the desired gradient.

4.2.2 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 (25a) with $2n_S$ sets of constraints (25b) and (25c) as

$$\mathcal{L} = f(\boldsymbol{\lambda}) + \sum_{m=1}^{n_S} \boldsymbol{\mu}_m^T (\mathbf{C}^T \mathbf{C} \mathbf{v}_m - \lambda_m \mathbf{v}_m) + \sum_{m=1}^{n_S} \nu_m (\mathbf{v}_m^T \mathbf{v}_m - 1) \quad (30)$$

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

Differentiation of (30) w.r.t b_r gives

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

which could be simplified to

$$\begin{aligned} \frac{\partial f}{\partial b_r} &= \sum_{m=1}^{n_S} \boldsymbol{\mu}_m^T \frac{\partial (\mathbf{C}^T \mathbf{C})}{\partial b_r} \mathbf{v}_m + \sum_{m=1}^{n_S} \left(\frac{\partial f}{\partial \lambda_m} - \boldsymbol{\mu}_m^T \mathbf{v}_m \right) \frac{\partial \lambda_m}{\partial b_r} \\ &+ \sum_{m=1}^{n_S} [\boldsymbol{\mu}_m^T (\mathbf{C}^T \mathbf{C} - \lambda_m \mathbf{I}_{n_S}) + 2\nu_m \mathbf{v}_m^T] \frac{\partial \mathbf{v}_m}{\partial b_r}. \end{aligned} \quad (31)$$

The key idea of the adjoint method is to circumvent the direct computation of the sensitivity of the state variables $\boldsymbol{\lambda}$ with respect to optimization variables \mathbf{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 (31), their coefficients in (31) 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

$$\begin{cases} \frac{\partial f}{\partial \lambda_m} - \boldsymbol{\mu}_m^T \mathbf{v}_m = 0, \\ \boldsymbol{\mu}_m^T (\mathbf{C}^T \mathbf{C} - \lambda_m \mathbf{I}_{n_S}) + 2\nu_m \mathbf{v}_m^T = \mathbf{0}. \end{cases} \quad (32)$$

The above equation can be rearranged into the following linear system

$$\begin{bmatrix} \mathbf{v}_m^T & 0 \\ \mathbf{C}^T \mathbf{C} - \lambda_m \mathbf{I}_{n_S} & 2\mathbf{v}_m \end{bmatrix} \begin{bmatrix} \boldsymbol{\mu}_m \\ \nu_m \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial \lambda_m} \\ \mathbf{0} \end{bmatrix} \quad (33)$$

from which the adjoint variables $\boldsymbol{\mu}_m$ and ν_m can be solved. The sensitivity in (31) can then be obtained through the following simplified expression

$$\frac{\partial f}{\partial b_r} = \sum_{m=1}^{n_S} \boldsymbol{\mu}_m^T \frac{\partial (\mathbf{C}^T \mathbf{C})}{\partial b_r} \mathbf{v}_m. \quad (34)$$

It is worth noting that, with the direct method, the linear system (29) 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 (33) is solved just once for every eigenvalue λ_m . Numerical examples in Section 5.2 are presented to demonstrate the accuracy and efficiency of the adjoint sensitivity.

4.3 Inequality constraints for optimization

4.3.1 Constraints aggregation

Based on the description in Section 3.2.2, we consider the two options: a) Jacobian B-spline (via Bézier implementation) (23); and b) sampled Jacobian (24). 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 Jacobian constraints for each shape into one single constraint to be applied as the optimization constraints in (25d), yielding totally $n_G = n_S - 1$ constraints for $n_S - 1$ shapes as

$$g_l(\mathbf{b}) = \frac{1}{K} \ln \left[\sum_{z=1}^{n_J} e^{-K J_z(\{\mathbf{b}\}_l)} \right] < 0, \quad l \in 1, \dots, n_S; \quad l \neq i_R \quad (35)$$

where J_z is original Jacobian positivity, being either $J_{s,t}^A$ in (23) for the Jacobian B-spline constraint or $J_{s,t}^B$ in (24) 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.

4.3.2 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 (35) can also be derived. The gradient of l -th inequality constraint w.r.t the r -th optimization variable b_r is

$$\frac{\partial g_l}{\partial b_r} = -\frac{1}{\sum_{z=1}^{n_J} e^{-KJ_z(\{\mathbf{b}\}_l)}} \sum_{z=1}^{n_J} e^{-KJ_z(\{\mathbf{b}\}_l)} \frac{\partial J_z(\{\mathbf{b}\}_l)}{\partial b_r} \quad (36)$$

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 (23) and (24).

5 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 synthetic data set is the “plane-bump” (3D) shapes. The use of real data sets, including 40 hands (2D) and 34 distal femur bones (3D), is to show the practicality of the proposed approach. An earlier version of the proposed approach has been used in optimizing correspondence among 6 aortic shapes [56].

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 (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.

5.1 3D synthetic data: Plane-bump

We 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

$$z = h e^{-\frac{(x-\mu)^2}{2\sigma^2}} e^{-\frac{(y-\nu)^2}{2\phi^2}} \quad (37)$$

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 Fig. 6(a). Such a plane-bump shape can be approximated by a B-spline surface and Fig. 6(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 Fig. 6(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. Fig. 6(b) shows the parameterization with fairly low distortion as indicated by its knot curves for Shape 1; the other three instances are similarly parameterized.

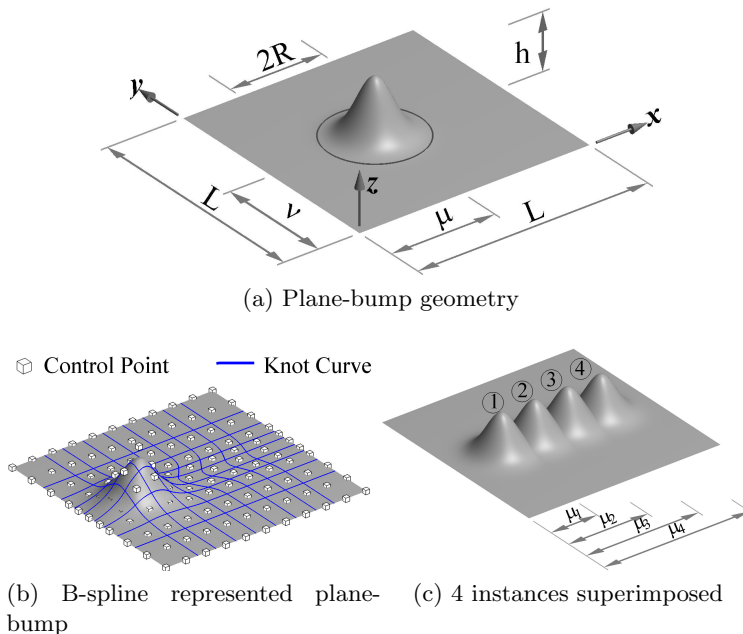


Figure 6: Plane-bump geometry and training set of 4 B-spline surfaces

5.1.1 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 Fig. 10(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 Fig. 10(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. Fig. 7(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).

5.1.2 Reparameterization by concatenation of Clamped Plate Splines

It’s possible to obtain the optimal correspondence through two reparameterization methods: concatenation of CPSs [29] 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 $N_{\text{OPT}} = 10^5$ 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 Fig. 7(b) approaches the desired trend shifting rightward along the x -axis very similar to Fig. 7(a). The reparameterization functions for Shape 2 and 3 are not shown but also exhibit similar trends.

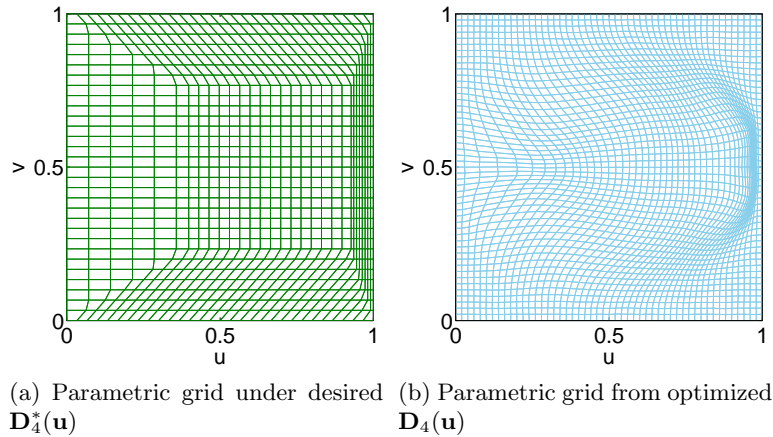


Figure 7: Desired regular landmark grid deformation and parametric grid under optimized $\mathbf{D}_4(\mathbf{u})$ with concatenation of Clamped Plate Spline warps.

5.1.3 Reparameterization via B-splines

Optimized reparameterization function The desired optimization obtained by the CPS concatenation reparameterization can also be achieved via the B-spline reparameterization approach. Fig. 8(d) shows the deformed landmark grid (originally a regular 51×51 grid) with 12×12 B-spline coefficients, which demonstrates similar motion in Fig. 7(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 Fig. 8(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.

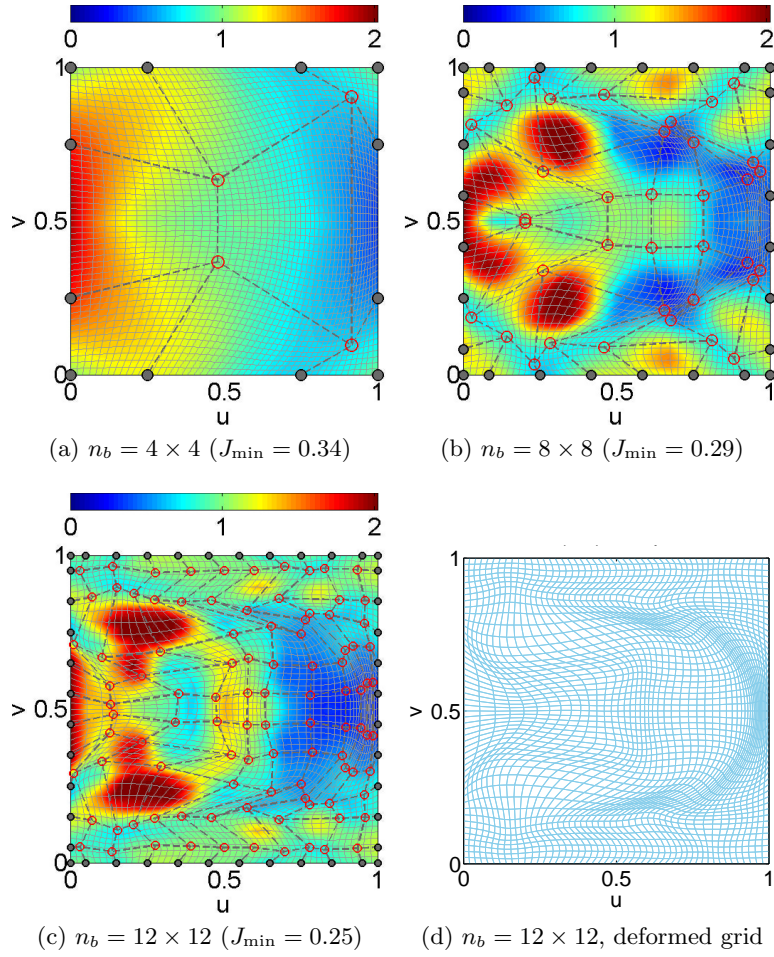


Figure 8: Optimized reparameterization $\mathbf{D}_4(\mathbf{u})$ for shape 4 obtained with different numbers of B-spline coefficients. The color field shows the Jacobian of $\mathbf{D}_4(\mathbf{u})$.

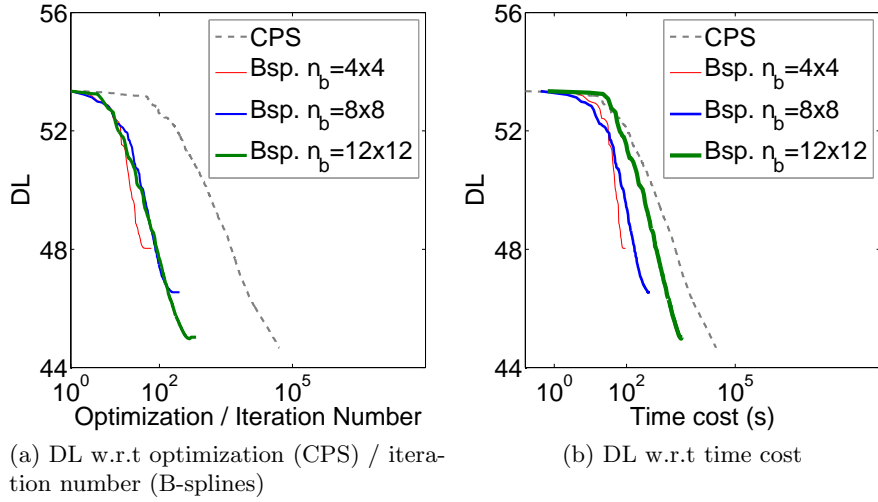


Figure 9: Time cost comparison: concatenation of CPS warps vs. direct reparameterization with B-splines.

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 Fig. 9(a) and w.r.t elapsed time in Fig. 9(b). It can be seen that in order to reduce the DL objective 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.

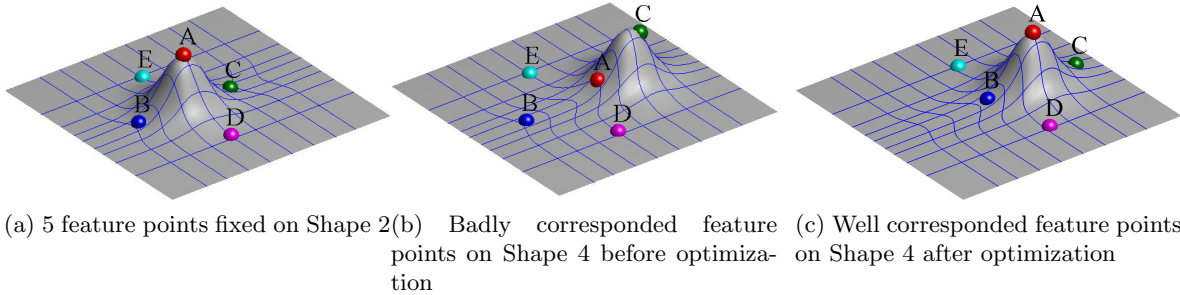


Figure 10: Correspondence before and after optimization with $n_b = 12 \times 12$ control points in reparameterization B-spline.

Correspondence improvement after optimization Fig. 10 demonstrates the significant correspondence improvement with optimization. Before optimization, the five points on Shape 4 shown in Fig. 10(b) corresponding to the five feature points fixed on the reference Shape 2 in Fig. 10(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 $n_b = 12 \times 12$ 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 Fig. 10(c).

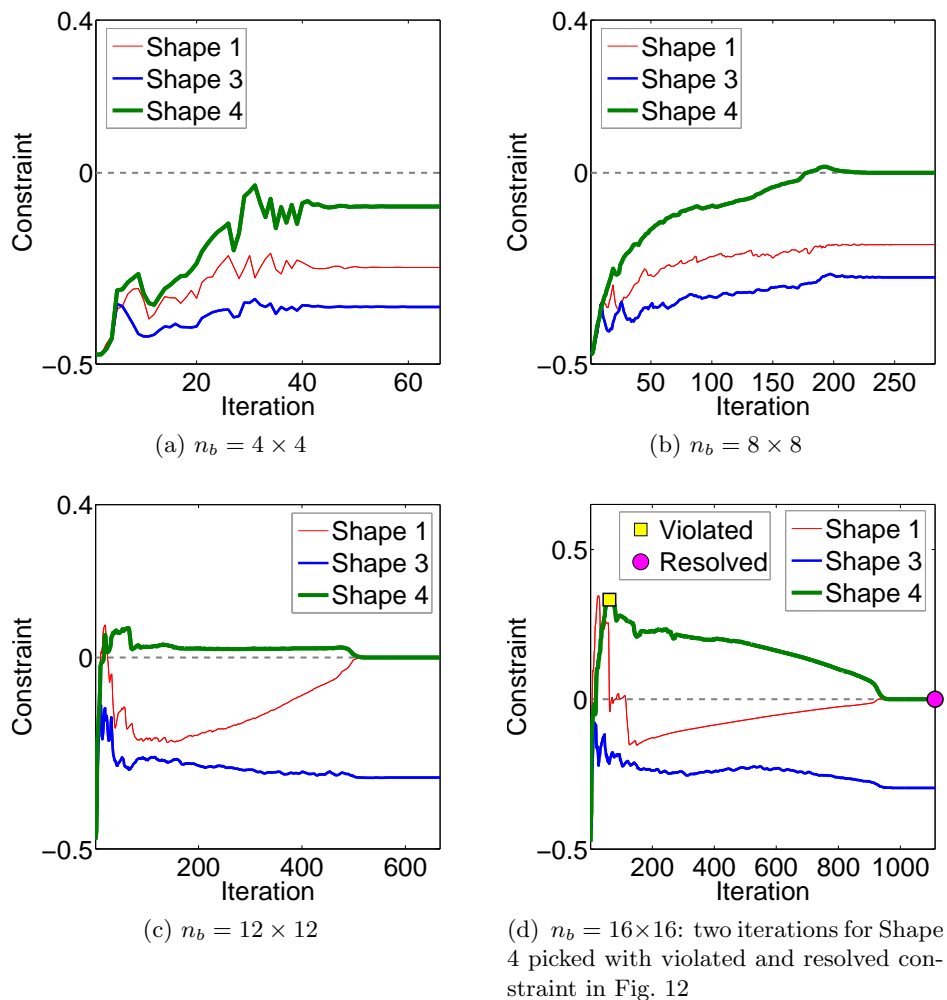
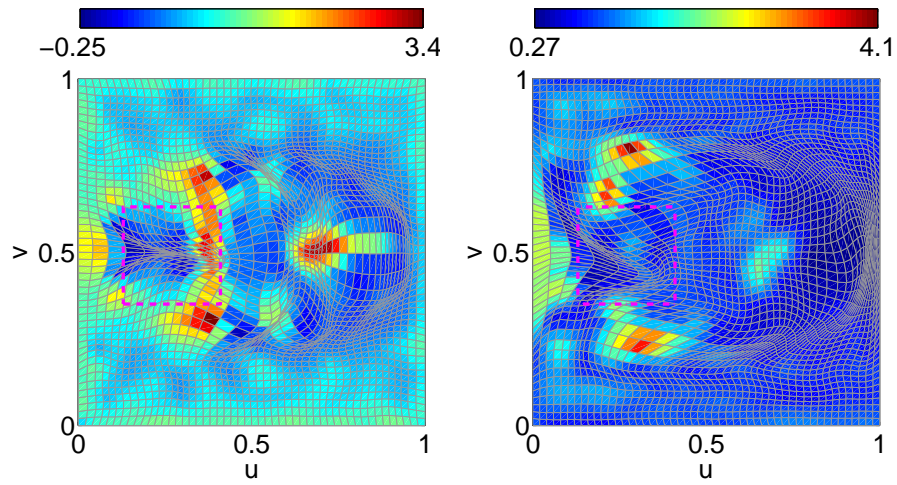


Figure 11: History of aggregated Jacobian constraint for the three non-reference shapes at four B-spline coefficient resolutions

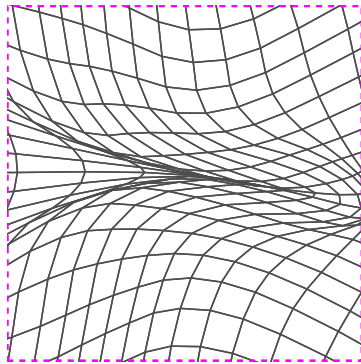
Constraint history and diffeomorphic conditions Fig. 11 shows the aggregated constraints of directly sampled Jacobians during the optimization iterations with different B-spline coefficients of 4×4 , 8×8 , 12×12 and 16×16 . At the coarse representation ($n_b = 4 \times 4$), the 3 aggregated constraints for shapes 2, 3 and 4 are not active throughout the optimization process as shown in Fig. 11(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 active (i.e. positive) as shown in Fig. 11(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

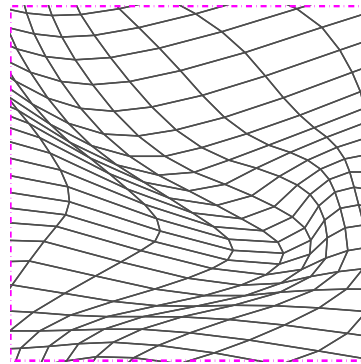


(a) The Jacobian with negative Jacobian at some landmark

(b) Positive Jacobian at all landmarks



(c) Self-intersection in (a)



(d) No self-intersection in (b)

Figure 12: Diffeomorphism through Jacobian constraints in shape 4 with 16×16 B-spline coefficients

16×16 . At resolution of 12×12 B-spline coefficients as shown in Fig. 8(c), although the aggregated Jacobian violates the constraint but the original Jacobian is still positive and no self-intersection exists. As the resolution 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 Fig. 11(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 Fig. 12(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 Fig. 12(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 $J_{\min}^{\text{raw}} = 0.27$. Fig. 12(c) and (d) shows the zoom-in around $\mathbf{u} = (0.2, 0.5)$ of Fig. 12(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 and sampled Jacobians In addition to using the aggregated sampled Jacobians J^B in (24) as optimization constraint $\{g_l(\mathbf{b})\}$ via (35), the aggregated Jacobian B-spline option with J^A as defined in (23) is also implemented for comparison. The time cost history between the two constraint options are shown in Fig. 13, where slightly better computational efficiency with J^A is observed.

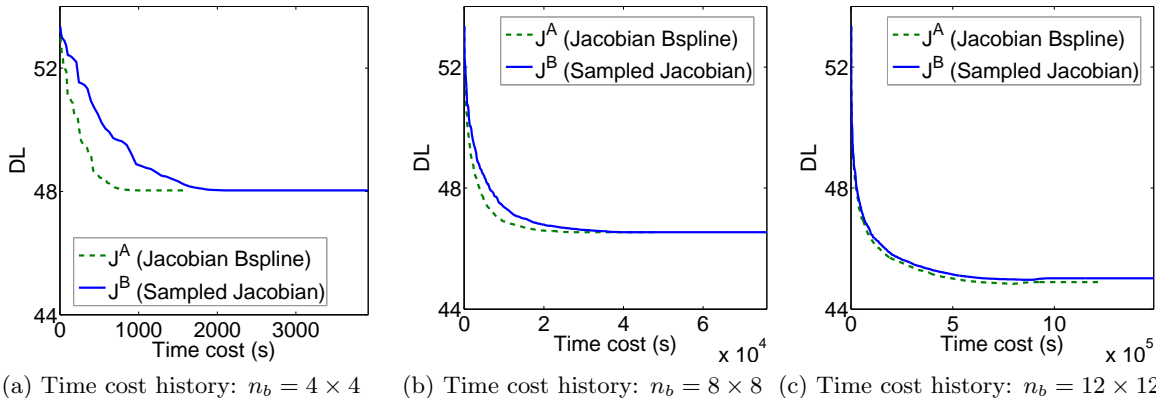


Figure 13: Comparison of optimization objective and constraint between constraint option A and B

In light of the ease of implementation using J^B and the small difference in the optimization results, the sampled Jacobian positivity constraint option is adopted for all the 3D experiments in this paper.

5.2 Computational analysis of analytical gradient

Section 4.2 introduces the analytical gradient formula in the form of direct and adjoint sensitivities. Now we use the 3D plane-bump example presented above to demonstrate the accuracy, correctness and efficiency of computing analytical gradient by the adjoint method.

5.2.1 Analytical gradient vs. numerical approximation

Within the context of the 3D plane-bump example in Section 5.1, the objective function’s analytical gradient $\frac{df(\mathbf{b})}{db_r}$ can be compared with its numerical approximation in terms of accuracy and time cost. The analytical equation refers to the direct sensitivity in (28) (29) and the adjoint sensitivity in (33) (34). The numerical approximation uses the finite difference method

$$\frac{df(\mathbf{b})}{db_r} \approx \frac{f(\mathbf{b} + h\mathbf{e}_r) - f(\mathbf{b})}{h}$$

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

Table 1 compares the gradients between direct and adjoint sensitivity along with the numerical approximation. The landmark resolution is $n_P = 51 \times 51$, and reparameterization B-spline resolution is $n_b = 4 \times 4$ as shown in Figure 14(a). There are total $n_D = 24$ optimization variables for the three non-reference shapes, thus the gradient of the cost function has 24 elements. The table shows 8 gradient values out of 24 for all three gradient evaluation options. It can be seen that the two analytical sensitivities are very close to each other (consistent through the 9th digit), and numerical approximation gives slightly off result (consistent only till the 3rd digit).

Table 1: Comparison of analytical (direct and adjoint) and numerical gradients.

$\frac{df}{db_r}$	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: Time comparison between analytical (direct and adjoint sensitivity) and numerical (finite difference) approach for gradient evaluation.

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

Table 2 shows the time cost comparison between analytical and numerical gradient computations. The landmark resolution is $n_P = 51 \times 51$. The reparametrization B-spline coefficients' resolution are $n_b = 4 \times 4$, $n_b = 8 \times 8$, $n_b = 12 \times 12$ and $n_b = 16 \times 16$. It can be seen that the adjoint approach is the fastest, the direct one is slightly slower and numerical approximation is the slowest; the computational efficiency advantage is more noticeable as more B-spline coefficients (i.e. more optimization variables) are used for reparameterization.

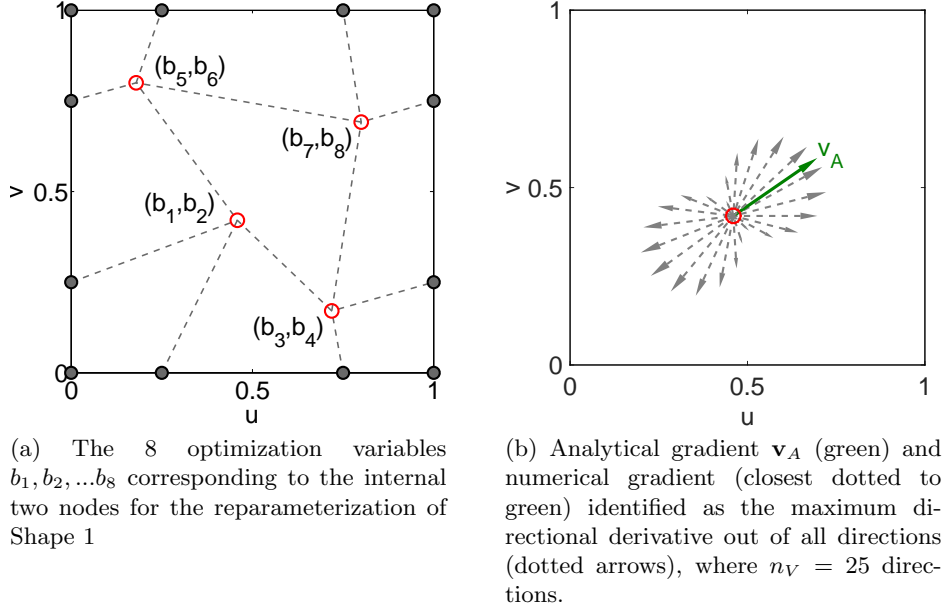


Figure 14: Optimization variables and directional derivatives

Besides finite difference based numerical approximation of the gradient, one can also compare analytical gradient with the approximated direction derivatives. Figure 14(a) shows the 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 for the other two shapes are randomly placed in the parameter domain for testing purpose. We compute the directional derivative of the 1st coefficient point (b_1, b_2) of Shape 1. We subdivide the full 360° range into n_V angle intervals. The directional derivative at \mathbf{p} along direction \mathbf{v}_i is approximated by finite difference as $\nabla_{\mathbf{v}_i} f(\mathbf{p}) \approx \frac{f(\mathbf{p} + h\mathbf{v}_i) - f(\mathbf{p})}{h}$. When the full circle is equally subdivided into $n_V = 25$ directions as shown in Figure 14(b), the directional derivatives are plotted in arrows pointing in their respective directions. The directional derivative with the largest magnitude is identified as the numerical gradient, which is fairly close to the analytical gradient computed by adjoint formula shown in solid green. Table 3 compares the analytical gradient with the maximum directional derivatives computed for different number of angle intervals. The row of $n_V = 25$ corresponds to the scenario in Figure 14(b). It can be seen that the numerical results tend toward the analytical gradient as number of angle intervals increases.

5.2.2 Direct vs. adjoint sensitivity

The key advantage of the adjoint method for computing the sensitivity over the direct sensitivity lies in the efficiency gain when there are a large number of optimization variables. This is because,

Table 3: Analytical gradient $\mathbf{v}_A = \left[\frac{df(\mathbf{b})}{db_1}, \frac{df(\mathbf{b})}{db_2} \right]^T$ 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

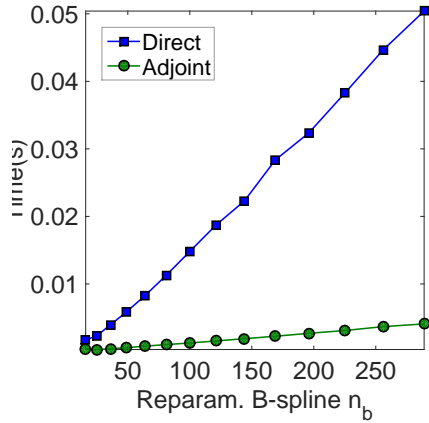
as shown in Section 4.2, the number of the linear system (29) solved in the direct sensitivity is proportional to the number of optimization variables and the number of eigenvalues. The number of linear system (33) solved in the adjoint sensitivity is proportional to the number of eigenvalues.

The number of optimization variables in optimizing shape correspondence depends on both the number of reparameterization coefficients per shape and the number of shapes used for building the SSM. Fig. 15(a) shows the influence of the number of reparameterization (B-spline) coefficients n_b on computation time when using direct and adjoint formulas. Here the landmark resolution is fixed at $n_P = 51 \times 51 = 2601$ for 3D plane-bump, and shape number is fixed at $n_S = 4$. It can be observed that the computational time cost for the adjoint sensitivity barely increases as the reparameterization coefficients increase while the cost for the direct sensitivity increases significantly. Fig. 15(b) shows the influence of shape population size (shape number) n_S on computation time when using direct and adjoint formulas. Here the reparameterization B-spline resolution is fixed at $n_b = 12 \times 12 = 144$, and the landmark resolution is fixed at $n_P = 51 \times 51 = 2601$. It can be observed that the adjoint formula is insensitive to the shape number and makes it very efficient for computing the correspondence sensitivity for a large shape population. In light of such significant efficiency advantage, the adjoint sensitivity is the method used in the subsequent numerical examples for correspondence optimization.

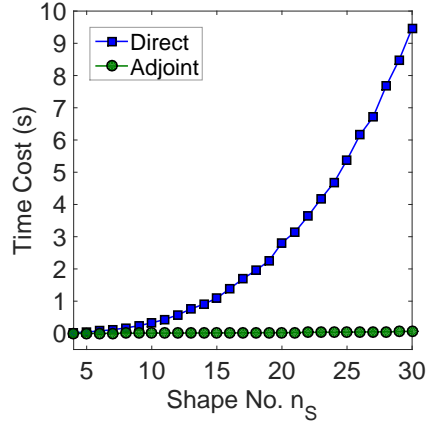
5.3 2D real data: Hand

5.3.1 Training set B-spline pre-processing

We test our correspondence optimization on a real data set from 40 raw images [57]. One example image is shown in Fig. 16(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 Fig. 16(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 represented hands are superimposed in Fig. 16(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 Fig. 16(d).



(a) Varying the number of reparameterization coefficients $n_b = n_{b1}n_{b2}$ per shape.



(b) Varying the number of shapes n_s , with $n_P = 2601$ ($n_{P1} = n_{P2} = 51$), $n_b = 144$ ($n_{b1} = n_{b2} = 12$) fixed

Figure 15: Computing time for direct sensitivity and adjoint sensitivity w.r.t the number of optimization variables: (a) increasing the number of reparameterization coefficients n_b per shape and (b) increasing the number of shapes.

5.3.2 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.

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 [57] 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 Fig. 16(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 Fig. 17(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 Fig. 17(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 Fig. 17(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

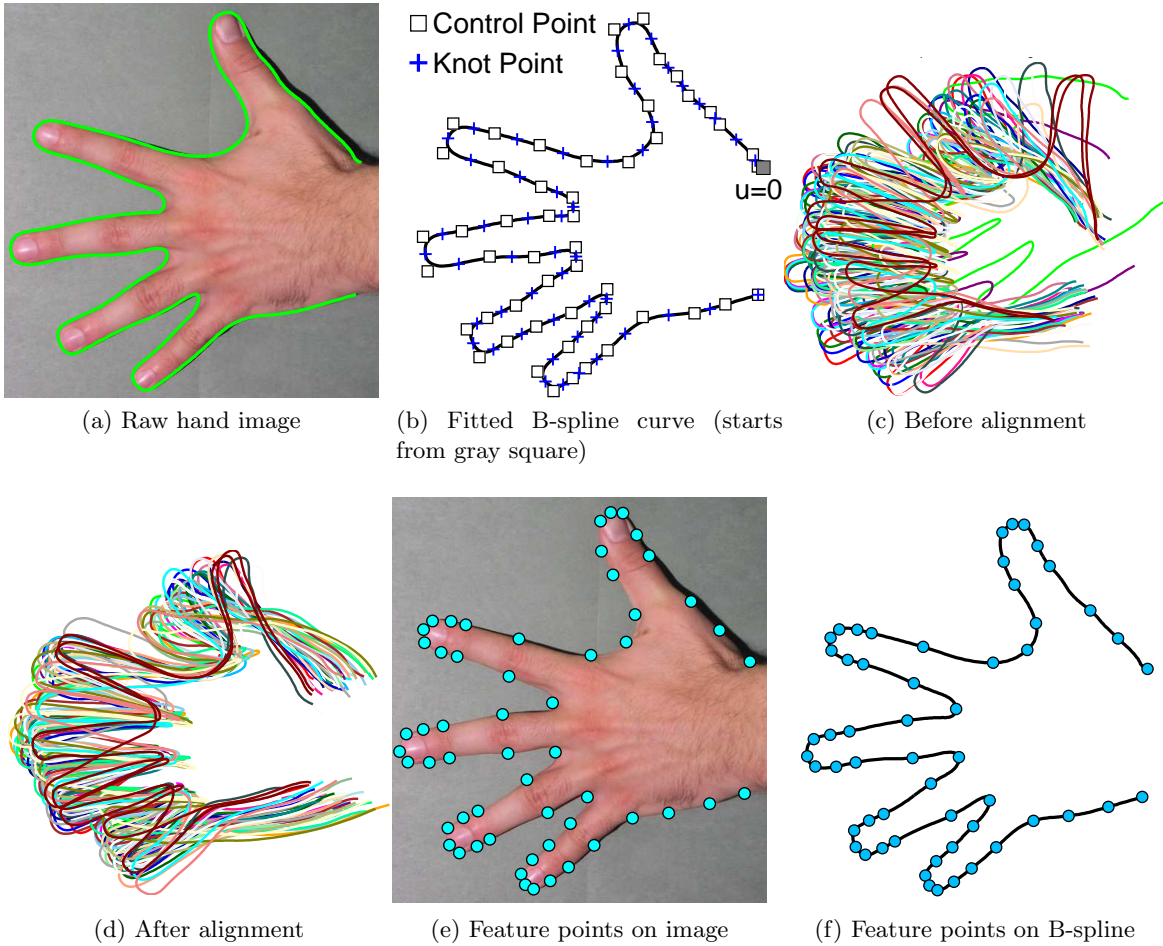


Figure 16: Pre-processing the training set of hand data.

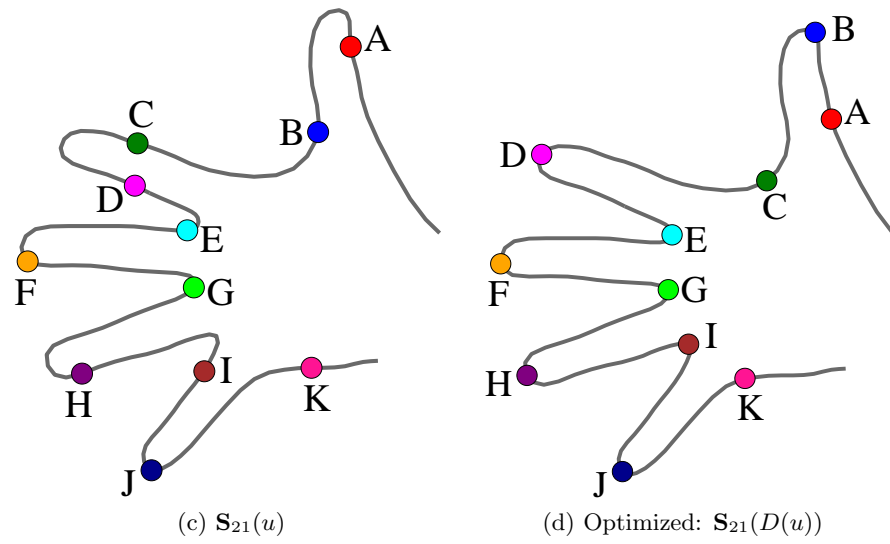
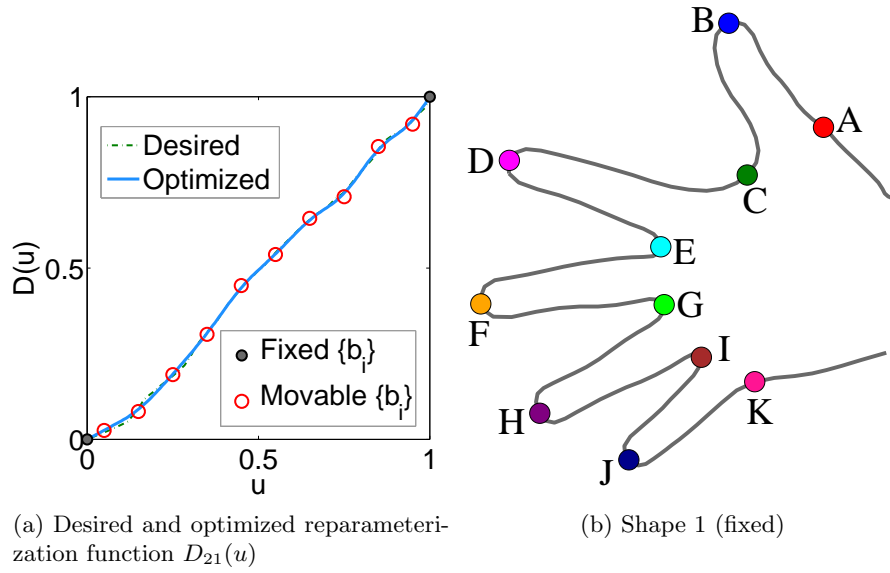


Figure 17: Feature points on Shape 21 before and after optimizing correspondence with respect to shape 1.

Fig. 17(d).

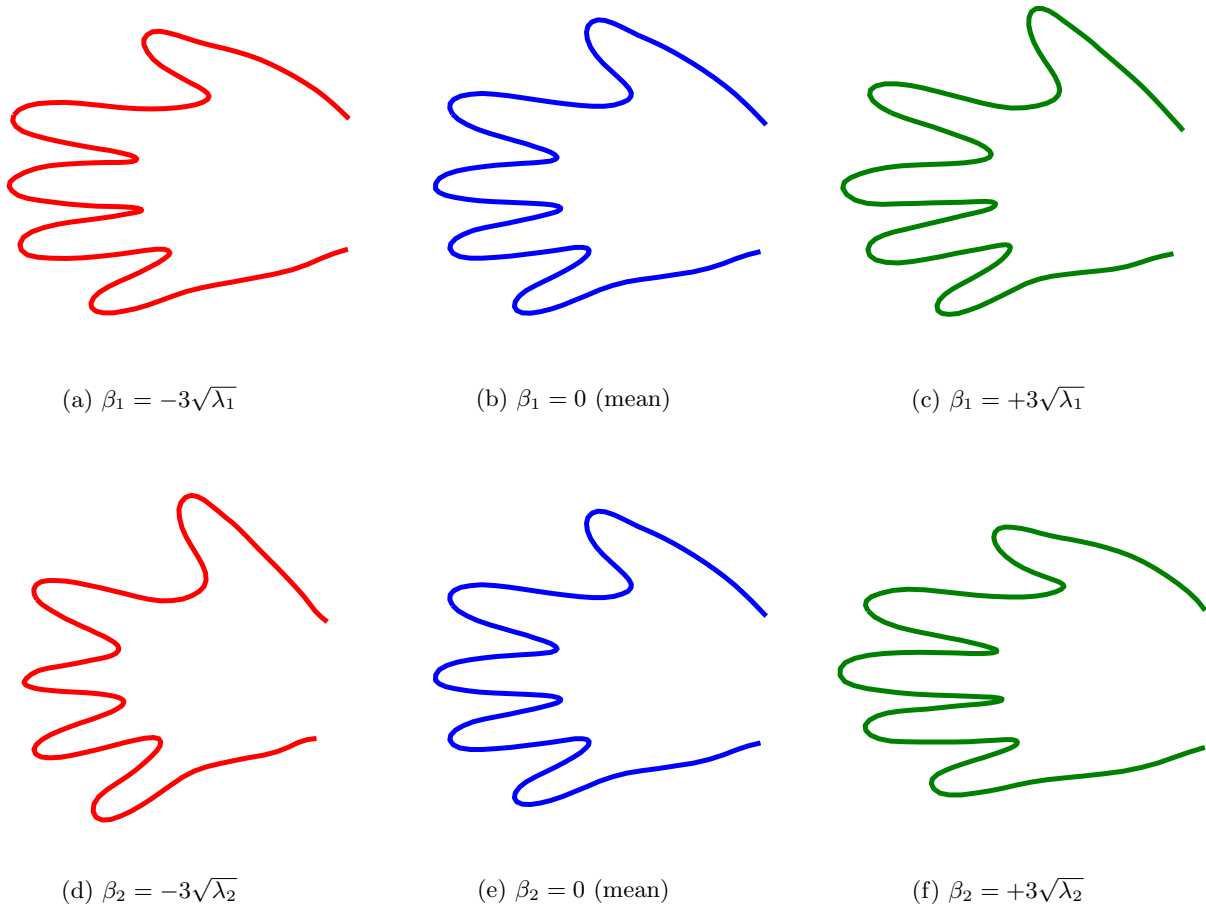


Figure 18: The first 2 modes of statistical model for hand data before optimization. $\lambda_1 = 723.2(31.48\%)$, $\lambda_2 = 467.7(20.36\%)$

Statistical modes improvement Fig. 18 and Fig. 19 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 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. Such differences in shape variations from the models are a direct consequence of the optimized correspondence from Fig. 17(c) to Fig. 17(d) as compared to Fig. 17(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

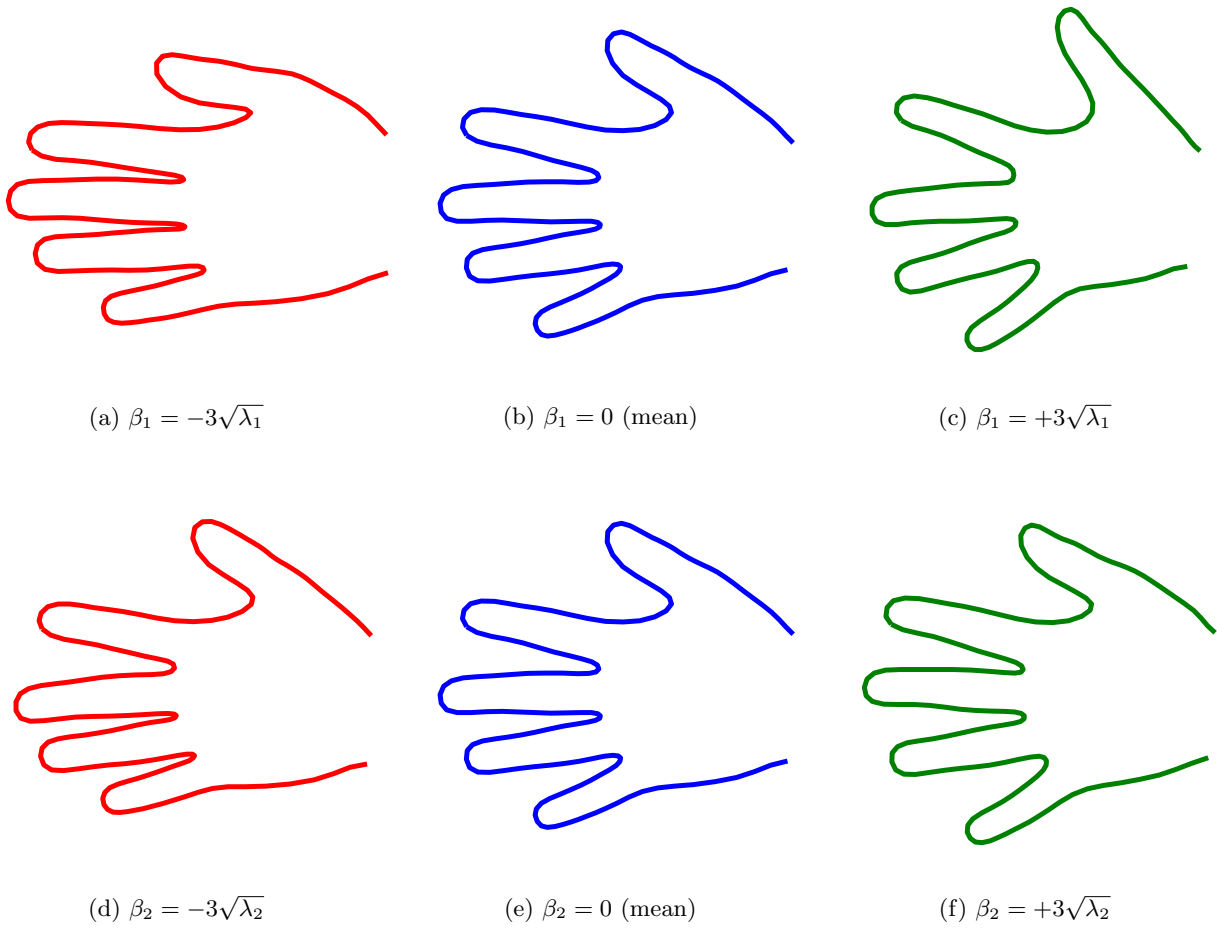


Figure 19: The first 2 modes of statistical model for hand data after optimization. $\lambda_1 = 438.4(50.24\%), \lambda_2 = 153.44(18.04\%)$

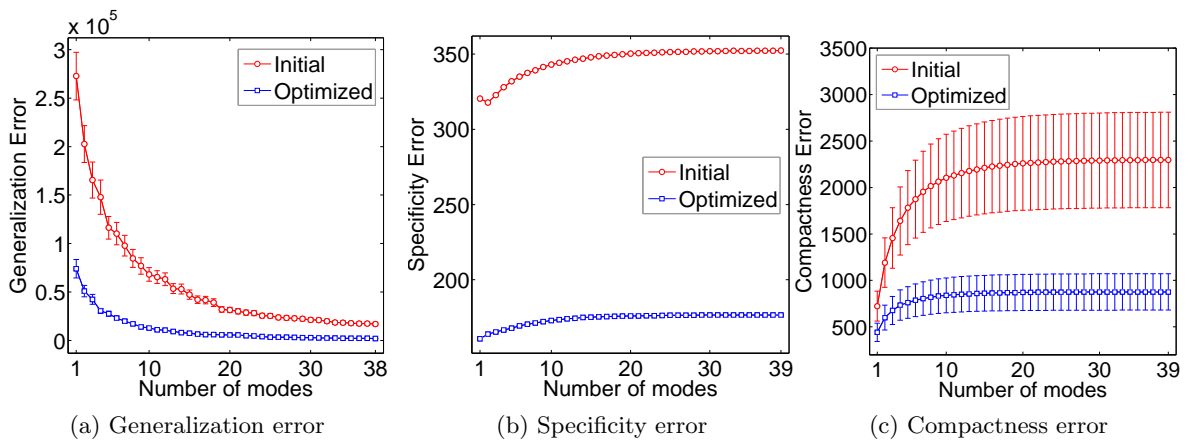


Figure 20: SSM quality before and after optimization

in [29] are computed. Fig. 20 compares the generalization ability, specificity and compactness error [29] 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 (9). Varying the number of modes n_m for the linear approximation, the leave-one-out generalization measure of SSM is given by

$$E_G(n_m) = \frac{1}{n_S} \sum_{i=1}^{n_S} \|\mathbf{x}_i - \tilde{\mathbf{x}}_i\|. \quad (38)$$

The standard error of generalization measure is defined by

$$\sigma_G(n_m) = \frac{\sigma}{\sqrt{n_S - 1}}. \quad (39)$$

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

A third useful quantitative measure is the compactness error defined by

$$E_C(n_m) = \sum_{m=1}^{n_m} \sqrt{\frac{2}{n_S}} \lambda_m \quad (40)$$

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 Fig. 21, 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.

5.4 3D real data: Distal femur

5.4.1 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. Fig. 22(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 [58] 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 [59] is then minimized and the obtained mesh parameterization is shown in Fig. 22(b).

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

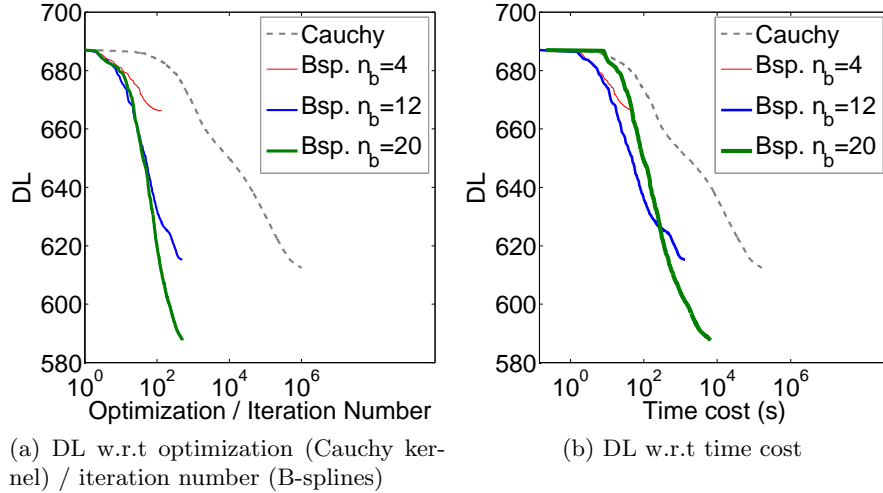


Figure 21: Time cost comparison: concatenation of Cauchy kernels vs. B-spline based reparameterization for correspondence optimization

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 Fig. 22(a) is now shown in Fig. 22(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.

5.4.2 Optimization via reparameterization B-spline

In the optimization, 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 Fig. 23(a), and the deformed parameterization grid under the optimized reparameterization function for Shape 34 is visualized in Fig. 23(b). The quantitative evaluation of the optimized SSM in generalization, specificity error and compactness are also performed and sizable improvements similar to Fig. 20 are observed.

Fig. 24 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.

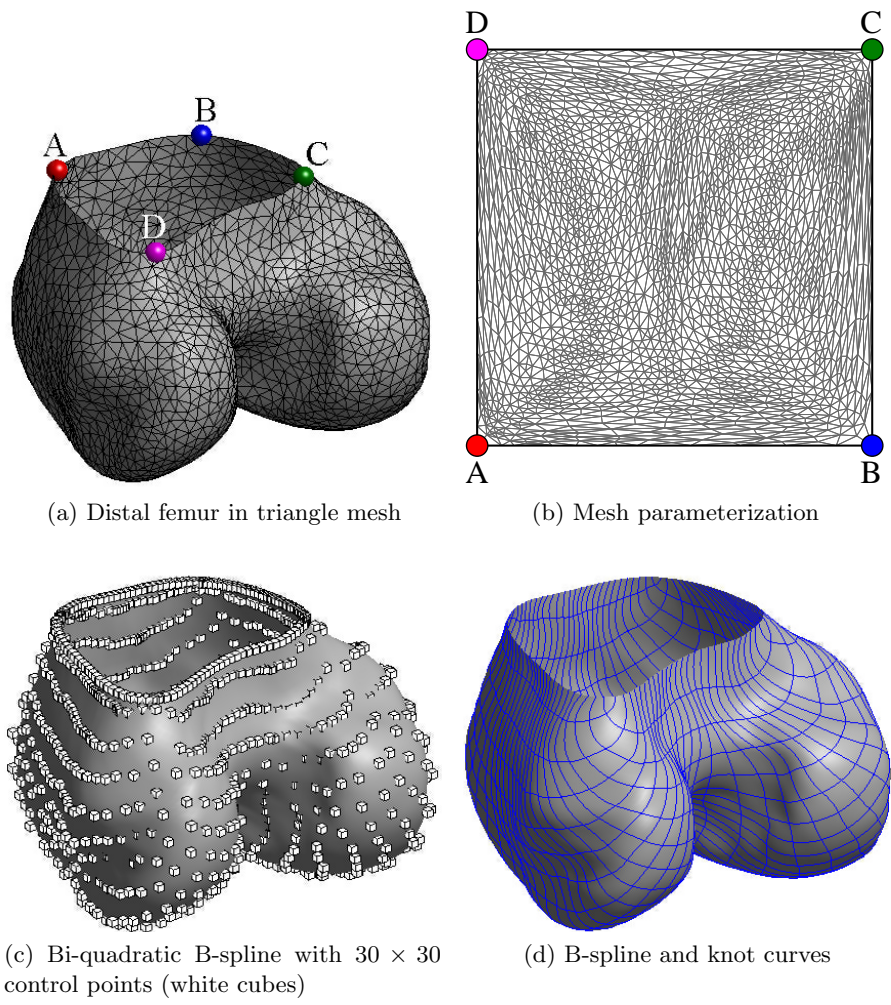


Figure 22: Pre-processing the distal femoral bone

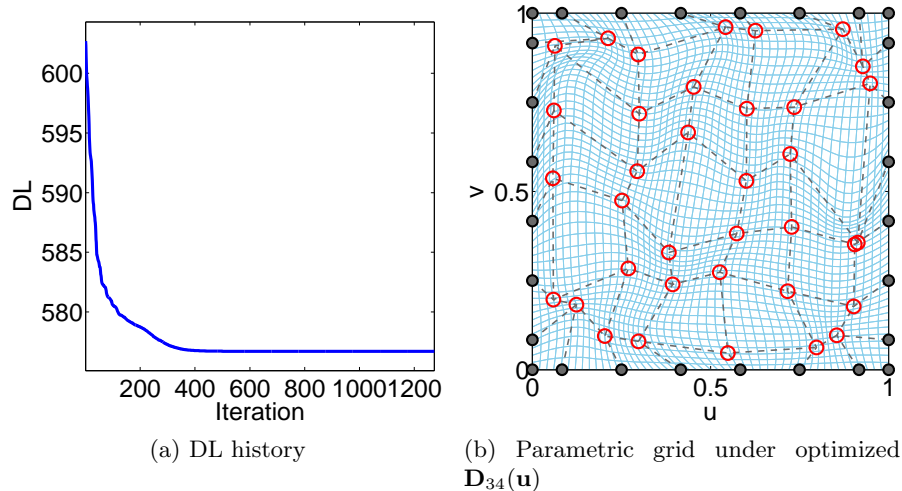


Figure 23: Optimization DL history and result of deformed grid of Shape 34 under the optimized reparameterization function with B-spline coefficients resolution of 8×8 .

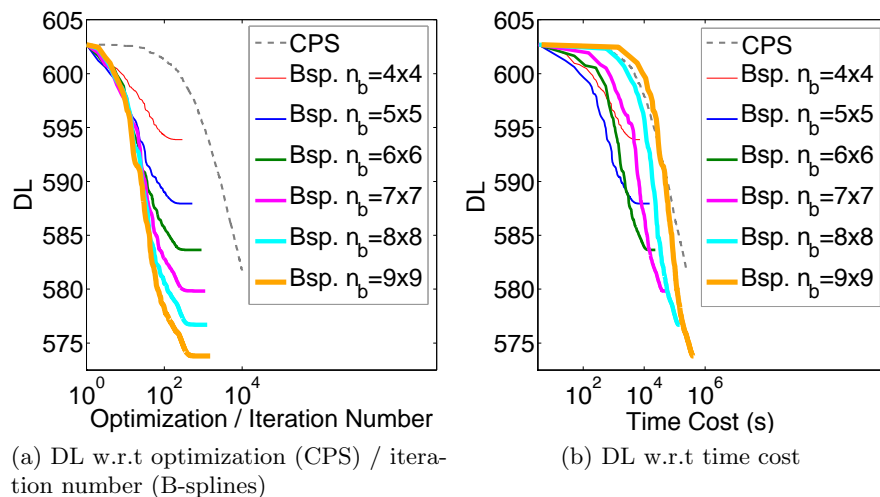


Figure 24: Time cost comparison: concatenations of CPS warps vs. direct B-spline based reparameterization for correspondence optimization for 34 shapes.

6 Conclusion

This paper introduces a novel reparameterization based 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 reparameterization methods such as Cauchy kernels and CPS warps for correspondence optimization.

In this paper, B-splines are used for both shape representation $\mathbf{S}(\mathbf{u})$ and reparameterization $\mathbf{D}(\mathbf{u})$ as in $\mathbf{S}(\mathbf{D}(\mathbf{u}))$. Such a combination makes it possible to directly use reparameterization variables to control the movement of landmarks on the surface shapes in a differentiable manner. Such differentiability makes it possible to develop the efficient gradient based approach for optimizing correspondence across a large shape population. However, B-spline based shape representation is limited to rectangular-like shapes. Alternative parameterization techniques that are applicable to shapes of complex topology are of interest for future research.

Acknowledgment

The authors acknowledge the financial support of AFOSR grant FA9550-12-1-0206 and NSF grants #0900597 and #1030347. We are also thankful for the data sources: 1) Active Appearance Models group at Technical University of Denmark for hand data; 2) Dr. Alejandro A Espinoza Orías at Rush University Medical Center for the femoral bone data.

References

- [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] Charlie CL Wang. Parameterization and parametric design of mannequins. *Computer-Aided Design*, 37(1):83–98, 2005.
- [12] Rhodri Huw Davies. *Learning shape: optimal models for analysing natural variability*. University of Manchester, 2002.
- [13] Ting Chen, Baba C Vemuri, Anand Rangarajan, and Stephan J Eisenschenk. Group-wise point-set registration using a novel cdf-based havrda-charvát divergence. *International journal of computer vision*, 86(1):111–124, 2010.
- [14] 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.
- [15] 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.
- [16] 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.
- [17] 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.
- [18] 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.
- [19] 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.

- [20] AC Kotcheff, Chris J Taylor, et al. Automatic construction of eigenshape models by direct optimization. *Medical Image Analysis*, 2(4):303, 1998.
- [21] 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.
- [22] 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.
- [23] Hans Henrik Thodberg. Minimum description length shape and appearance models. In *Information Processing in Medical Imaging*, pages 51–62. Springer, 2003.
- [24] Hans Henrik Thodberg and Hildur Olafsdottir. Adding curvature to minimum description length shape models. In *British Machine Vision Conference, BMVC*, 2003.
- [25] 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.
- [26] 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.
- [27] Jiri Hladuvka and Katja Bühler. Mdl spline models: Gradient and polynomial reparameterizations. In *17th British Machine Vision Conference*, pages 869–878, 2005.
- [28] 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.
- [29] Rhodri Davies, Carole Twining, and Chris J Taylor. *Statistical models of shape: Optimisation and evaluation*. Springerverlag London, 2008.
- [30] Colin Goodall. Procrustes methods in the statistical analysis of shape. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 285–339, 1991.
- [31] 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.
- [32] Ian Jolliffe. *Principal component analysis*. Wiley Online Library, 2005.
- [33] Les A Piegl and Wayne Tiller. *The NURBS book*. Springer Verlag, 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] Kang Li, Xiaoping Qian, Caitlin Martin, and Wei Sun. Toward patient-specific computational study of aortic diseases: A population based shape modeling approach. In *Proceedings of the ASME 2014 International Design and Engineering Technical Conferences & Computers and Information in Engineering Conferenc*, pages 428–441. ASME, 2014.
- [57] 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.
- [58] William T Tutte. How to draw a graph. *Proc. London Math. Soc*, 13(3):743–768, 1963.
- [59] 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.