DIFFEOMORPHIC SHAPE MATCHING BASED ON AN OPERATOR SPLITTING METHOD

A Dissertation Presented to

the Faculty of the Department of Mathematics

University of Houston

In Partial Fulfillment

of the Requirements for the Degree

Doctor of Philosophy

By

Peng Zhang

August 2019

DIFFEOMORPHIC SHAPE MATCHING BASED ON AN OPERATOR SPLITTING METHOD

Peng Zhang

APPROVED:

Prof. Robert Azencott
Dept. of Mathematics

Prof. Jiwen He
Dept. of Mathematics

Prof. Andreas Mang
Dept. of Mathematics

Dr. William A. Zoghbi
Cardiovascular Imaging Department
Houston Methodist
DeBakey Heart and Vascular Center

Dean, College of Natural Sciences and Mathematics

Acknowledgements

I am very lucky to do research in image analysis with optimization. This dissertation would not have been possible without collaborative efforts and support from a lot of people.

Firstly, I would like to thank both my advisers Dr. Robert Azencott and Dr. Jiwen He for introducing me to this medical project in cooperating with Dr. William A. Zoghbi. Dr. Azencott is talented and enthusiastic. He has lots of ideas and powerful magic to change impossible to possible. He is positive all the time and calms me down when I am confused. Dr. He is very good at computing and numeric analysis. He introduced the operator splitting method into our program, which is very efficient. He has a very accurate foresight for our model and always led me right towards it. Dr. Zoghbi is very nice. I really respect his insistence on developing echocardiographic and mathematical techniques to evaluate valvular dynamics. He has funded me as a research assistant for three years. Without their guidance, support, and patience, I would not have made this far.

I also want to thank Dr. Andreas Mang for organizing our group meeting every week and sharing his ideas, Dr. Shanyu Ji for enrolling me in the University of Houston and taking care of my life, Dr. Edward Kao for teaching me five courses. I would like to thank Dr. Aarti Jajoo and Dr. Yue Qin for sharing their experience and code.

Last but not least, I would like to mention the great support from my parents and friends, especially my wife Xi Chen, without which certainly it would not have

been possible to reach this goal.

DIFFEOMORPHIC SHAPE MATCHING BASED ON AN OPERATOR SPLITTING METHOD

An Abstract of a Dissertation

Presented to

the Faculty of the Department of Mathematics

University of Houston

In Partial Fulfillment

of the Requirements for the Degree

Doctor of Philosophy

By

Peng Zhang

August 2019

Abstract

In order to study the valvular dynamic relying on a sequence of non-invasive echocardiographic 3D images of the mitral valve, one main goal is the reconstruction of shape deformations. In this dissertation, we use a spline technique to generate a static model of the mitral valve, which contains around 2000 points per snapshot. We find the optimal diffeomorphic matching of these mitral valves using an operator splitting method. To make the problem computationally accessible, we consider reproducing kernel Hilbert spaces using Gaussian kernels and weighted sums of Dirac measures. The objective function includes the kinetic energy of the velocity and a shape matching term. In order to efficiently optimize it, we apply an operator splitting method. Then based on the optimal diffeomorphism, the strain value of every point around the surface is calculated through a local surface strain tensor. A machine learning approach based on support vector machines is used to automatically classify patients into several subgroups.

Contents

List of Figures

List of Tables

CHAPTER 1

Introduction

This is a long term project. We are thankful to Houston Methodist Hospital cardiology team led by Doctor William Zoghbi. Without their help, this dissertation would not have been possible. The task in automated 3D image registration is the reconstruction of a 3D movie on the basis of image snapshots S_j, acquired at successive times t_j, $0 \leq j \leq q$. This can be achieved by computing a time-dependent family of R^3 diffeomorphisms F_t such that $F_{t_0}(S_0) = S_0$ and $F_{t_j}(S_0)$ is as close to S_j, $1 \leq j \leq q$, as possible in terms of an appropriately chosen matching quality criterion. G. Dupuis, U. Grenander, M. Miller, A. Trouvé, M. F. Beg and L. Younes have published several fundamental research papers in this field [16, 40, 42, 8, 20, 24, 19, 16].

In the paper [2], R. Azencott, R. Glowinski, J. He, and R. Hoppe have applied diffeomorphic shape matching to movie reconstruction from a finite number of snapshots. In collaboration with R. Azencott, J. He, W. Zoghbi (MD), S.Little (MD), S. Ben Zekry (MD), several PhD students and Post Docs (A. Jajoo, J. Freeman, Y. Qin) have paved the way for the strain study of live mitral valves in human patients [49, 50, 48, 30, 18, 47]. In the work of A. Jajoo [30], a gradient method with Armijo line-search is used to compute a diffeomorphic registration flow minimizing an adequate cost function. Y. Qin [47] applies a second order gradient descent to the same problem to construct a minimizing diffeomorphic matching based on Bellman's optimality principle and she develops a dynamic programming algorithm, which works well when the number of data points per discretized snapshot is less than 400.

This work involves a patient specific discretized modeling of the mitral valve dynamics with a live human heart. With the help of K. C. El Tallawi (MD from Houston Medthodist Hospital) using a TOMTEC segmentation software, a finite sequence of discretized 3D snapshots of mitral valve is generated, like Figure 2.1. However this TOMTEC software only implements successive but separate segmentations of each image frame, so that actual tracking of the mitral valve dynamics is definitely not provided by the software and part of our algorithmic work was to actually develop a fast and accurate diffeomorphic registration of successive image snapshots. Using the finite grids generated by TOMTEC sofware for each 3D snapshot of the mitral valve, we modelize the dynamic deformation of the mitral valve by a diffeomorphic flow to be computed by minimizing an adequate cost function. Then, the strain values distribution on the mitral valve leaflets is computed and studied. This helps

us to evaluate which mitral valve regions have high or low strain.

The thesis is organized as follows. Chapter 2 focuses on the construction of the mitral valve. To do this, we reparameterize the finite grids provided by TOMTEC software, using cubic spline interpolation and thin plate splines. Chapter 3 introduces the mathematical foundation of the valve tracking problem. Chapter 4 provides a detailed study of the splitting method, which includes a quadratic control step and a single period proximal step. Chapter 5 provides the test results of our approach. To improve the matching of boundaries for mitral valve leaflets, we introduce weights in our formulation to better match the boundary points. This is explained in Chapter 6. In the next chapter, a comparison of the operator splitting model with a second order variational model [47] developed by Y. Qin is illustrated. Our new algorithmics based on operator splitting is much more efficient and faster. It is very robust and enables us to deal with large data sets. Chapter 8 handles one of the main goals of our research project: strain values calculation and analysis. We use a novel method to quantitatively analyze and compare the strain values distribution computed at around 2000 points of the mitral valve surface. In Chapter 9, the regional strain values distributions will be studied for several anatomic regions of the mitral valve surface. Finally, automatic classification for various clinically defined groups of patients is implemented using support vector machines. The accuracy achieved by our SVM approach is about 80%.

CHAPTER 2

Mitral Valve Modeling

The mitral valve is an important valve in the human heart. When it opens, the blood will go through from the left atrium to the left ventricle. The mitral valve has two leaflets, an anterior leaflet and a posterior leaflet. In a normal patient, the anterior leaflet and posterior leaflet will close to prevent the blood backflow into the atrium. In mitral valve prolapse, the valve leaflets balloon upward as the ventricle contracts due to the mitral valve leaflets' abnormal size or damaged mitral valve tissues. For most patients with mitral valve prolapse, the root cause of the problem is unknown. Nearly 8 million people in the US have mitral valve prolapse. The most severe situation, named regurgitation, is when the valve leaflets do not properly close,

forcing blood back into the atrium.

For each 3D image snapshot of a given patient's mitral valve, the TOMTEC software is used interactively by K. C. El Tallawi (MD at Houston Methodist Hospital). He provides us with a finite 3D grid of points discretizing the mitral valve surface. In general, there are around 8 to 14 three dimensional image frames for each patient. For each three dimensional frame, two leaflets are included, the anterior leaflet (820 points) and the posterior leaflet (820 points). This is shown in Figure 2.1. The anterior leaflet (magenta) and posterior leaflet (cyan) have 40 intersection points lying on two commissures curves (black). The annulus (blue), the anterior coaptation curve (red), the posterior coaptation curve (green) respectively have 80, 40 and 40 points.

Based on these grid data, we construct a uniform mesh which is better suited for our optimization. This means the size of every grid will be almost the same. We will separate anterior leaflet and posterior leaflet, and use principal component analysis for pre-alignment, then generate a new model using the spline interpolation techniques.

2.1 Principal Component Analysis

Suppose we have $L+1$ frames per sample. For the image frame acquired at time t, denote by $S^t \equiv \{PT_i^{(t)} | 1 \leq i \leq N_t\}$ the set of frame surface points, introduce the subset $S^t_{ant} \equiv \{PA^t_i | 1 \leq i \leq N_a\}$ corresponding to the anterior leaflet and the subset $S^t_{pos} \equiv \{PP^t_i | 1 \leq i \leq N_p\}$ discretizing the posterior leaflet, where $0 \leq t \leq L$; typically $N_t = 1600$, $N_a = 820$ and $N_p = 820$.

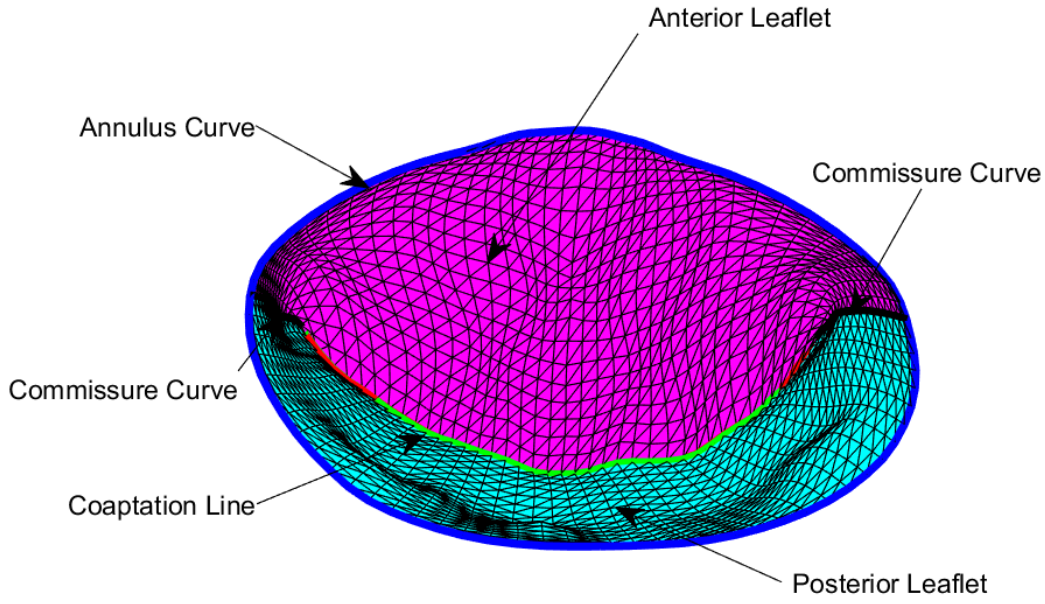


Figure 2.1: Mitral valve: when the valve is closed, the anterior coaptation line will meet the posterior coaptation line. Anterior leaflet (magenta), posterior leaflet (cyan), annulus (blue), commissure curve (black), anterior coaptation line (red), posterior coaptation line (green).

For each leaflet, principal component analysis (PCA) is applied for pre-alignment of the leaflet surface. We choose the anterior leaflet S^1_{ant} as an example to illustrate how it works. For simpler notation, we ignore the index t, i.e., $S_{ant} \equiv \{PA_i | 1 \leq i \leq N_a\}$. In 3D space, each point is represented by $PA_i \equiv (x_i, y_i, z_i)$. The PCA is done as follows. Step 1, find the center point $PC \equiv (c_1, c_2, c_3)$, which is the mean value of each coordinate. Step 2, center the data. For each point PA_i, the new coordinates are $PA'_i \equiv (x'_i, y'_i, z'_i) \equiv (x_i - c_1, y_i - c_2, z_i - c_3)$. Step 3, use the covariance matrix

(Σ) of the centered data; the principal components (e_1'', e_2'', e_3'') are computed based on a singular value decomposition. The vectors (e_1'', e_2'', e_3'') are the eigenvectors of Σ corresponding to the eigenvalues $\lambda_1, \lambda_2, \lambda_3$. Step 4, transfer the points into a new basis. We get $S''_{ant} \equiv \{PA_i''|1 \leq i \leq N_a\} \equiv \{(x_i'', y_i'', z_i'')|1 \leq i \leq N_a\}$ as new coordinates [29].

The purpose of pre-alignment is trying to construct the surface function $g(x'', y'')$ for a given mesh S''. This is what we discuss next.

2.2 Fitting Leaflet Boundaries By Cubic Smoothing Spline

The boundary of the anterior leaflet contains four pieces: part of the annulus, two commissures, and an anterior coaptation line. They will be fitted separately. To simplify the notation, we denote the new coordinates by xyz. The parametric curve $(x(\tau), y(\tau), z(\tau))$ is denoted by S_c.

We focus on finding a spline function $f(\tau) \equiv f(x(\tau), y(\tau))$ to fit the data S_c.

Consider the following problem: Among all the function f with two continuous derivatives, find the one that minimizes the penalized residual sum of squares

$$PRSS(f, p) = p \sum_{i=1}^{m} |z_i - f(x_i, y_i)|^2 + (1 - p) \int |D^2 f(\tau)|^2 d\tau, \tag{2.1}$$

where m is the number of curve points. The smoothing parameter p is chosen empirically to offer a balance between the error measure, which measures closeness to the data,

$$RSS(f) = \sum_{i=1}^{m} (z_i - f(x_i, y_i))^2,$$

and the roughness penalty functional, which penalizes the curvature of the function f

$$J(f) = \int |D^2 f(\tau)|^2 d\tau.$$

Two special cases are:

- $p = 1$: no penalty is imposed, the solution can be any function that interpolates the data.
- $p = 0$: the solution is the least square line fit to the data [25].

Remarkably, it can be shown that equation (2.1) has an explicit, finite-dimensional, unique minimizer, which is a natural cubic spline with knots at unique values of (x_i, y_i), $i = 1, ..., m$ [25].

In addition, to control the level of fit at different data points, weights w_i are added to the smoothing spline. The problem becomes

$$p \sum_{i=0}^{m} w_i |z_i - f(x_i, y_i)|^2 + (1 - p) \int_a^b |D^2 f(\tau)|^2 d\tau.$$

In practice, we use the MATLAB function csaps of the Curve Fitting Toolbox to get the cubic smoothing splines [6].

The fitted smoothing spline $\hat{f}(\tau)$ is computed as an approximation function to predict new values.

2.3 Fitting Leaflet Surface By Thin-plate Spline

For each leaflet surface, we implement smoothing by thin plate splines.

We want to fit the leaflet surface by a single Cartesian equation denoted as

$$z = g(x, y).$$

The idea is to set up the variational problem

$$\min_g p \sum_{i=1}^{m} |z_i - g(x_i, y_i)|^2 + (1-p)J[g],$$

where the roughness penalty functional on R^2 is

$$J[g] = \int\int_{R^2} [(\frac{\partial^2 g(x,y)}{\partial x \partial x}) + 2(\frac{\partial^2 g(x,y)}{\partial x \partial y}) + (\frac{\partial^2 g(x,y)}{\partial y \partial y})] dxdy.$$

Solving this variational problem leads to a smooth two dimensional surface, which is a thin-plate spline. Similarly, for $p = 1$, the solution approaches an interpolating function. For $p = 0$, the solution is the least square fit. We use the MATLAB function tpaps in the curve fitting tool box [6] to get the thin-plate spline, which is

of the form

$$g(x,y) = \beta_0 + \beta_1 x + \beta_2 y + \sum_{j=1}^{m} \alpha_j h_j(x,y),$$

where $h_j(x,y) = ||(x,y) - (x_j,y_j)||^2 \log ||(x,y) - (x_j,y_j)||^2$. These h_j are examples of radial basis functions. Using the existing data S, we can estimate the coefficients β_0, β_1, β_2, and α_j, $1 \leq j \leq m$.

The fitted smoothing spline $\hat{g}(x,y)$ is a smooth interpolation function for the leaflet surface.

We start by creating a regular grid in the (x,y) plane. Let $a1$, $a2$ represent the lower bound and upper bound of the first coordinate of the surface S_{ant} and $b1$, $b2$ the bounds for the second coordinate, respectively. Along the x coordinate, we partition the line from $a1$ to $a2$ equally into N_r pieces. The length for each piece is $h \equiv (a2 - a1)/N_r$. Then, the spline knots for x coordinate are $\{a1, a1 + h, a1 + 2h, \cdots, a1 + N_r h\}$.

Along the y coordinate, we use the same length h for each curve segment. Choosing $b1$ as the start knot, add h to the former knot to get the next knot, until the knot at the upper bound $b2$. Suppose we have M pieces, the knots for y coordinate will be $\{b1, b1 + h, b1 + 2h, \cdots, b1 + Mh\}$.

Using the knots of the x, y coordinates, we generate a mesh in the (x,y) plane, which is displayed in Figure 2.2.

Using the thin-plate spline, the surface equation is approximated by $z = \hat{g}(x,y)$.

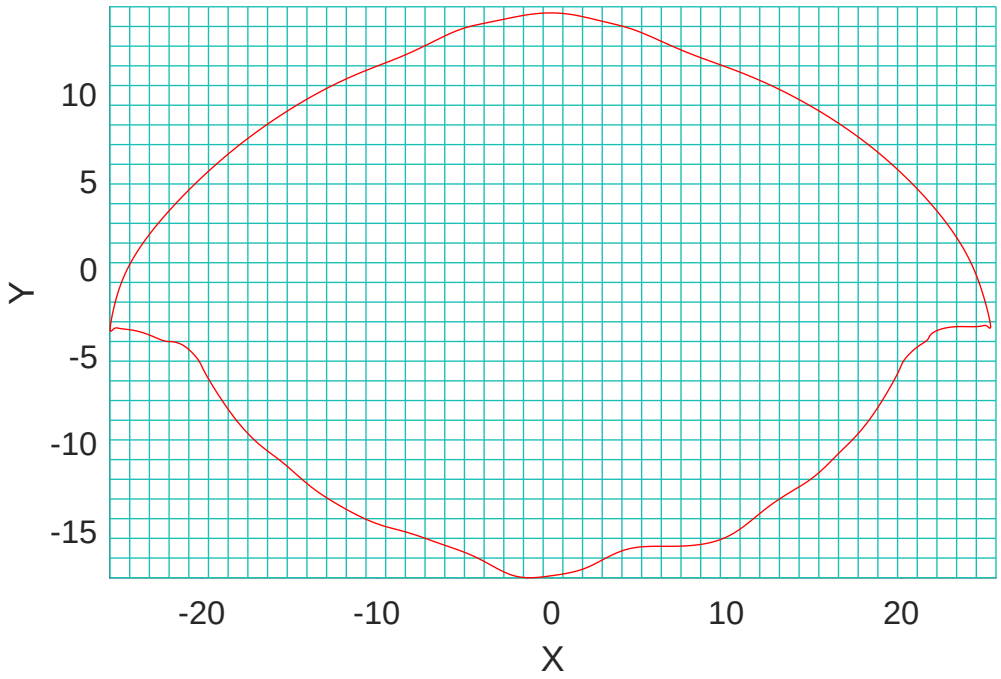


Figure 2.2: Construction of base plane ($N_r = 46, h = 1.125, M = 30$), red line is the boundary

2.4 Combining Leaflet Surface and Boundary

For each leaflet, define its interior points as the leaflet points which are inside the boundary and have a distance to the boundary larger than $3h/4$.

Denote the starting point of a leaflet boundary curve by p_b, and its end point by p_e. Set $b_0 \equiv p_b$, then find the boundary point b_1 which is at a distance almost h from the initial point b_0. After that we similarly find b_2 at distance h from b_1. In the same way, we can find $b_3, \cdots, b_n$ with b_n close to the curve end point p_e. If $||b_n - p_e|| > h/3$, we will add b_n to the discretized boundary curve which will be $\{b_0, b_1, \cdots, b_n, p_e\}$. Otherwise, we will pick p_e and disregard b_n; the discretized boundary is then $\{b_0, b_1, \cdots, b_{n-1}, p_e\}$.

Figure 2.3, 2.4, 2.5 show that the differences between the new grid points just constructed and the original TOMTEC grid points data are fairly small.

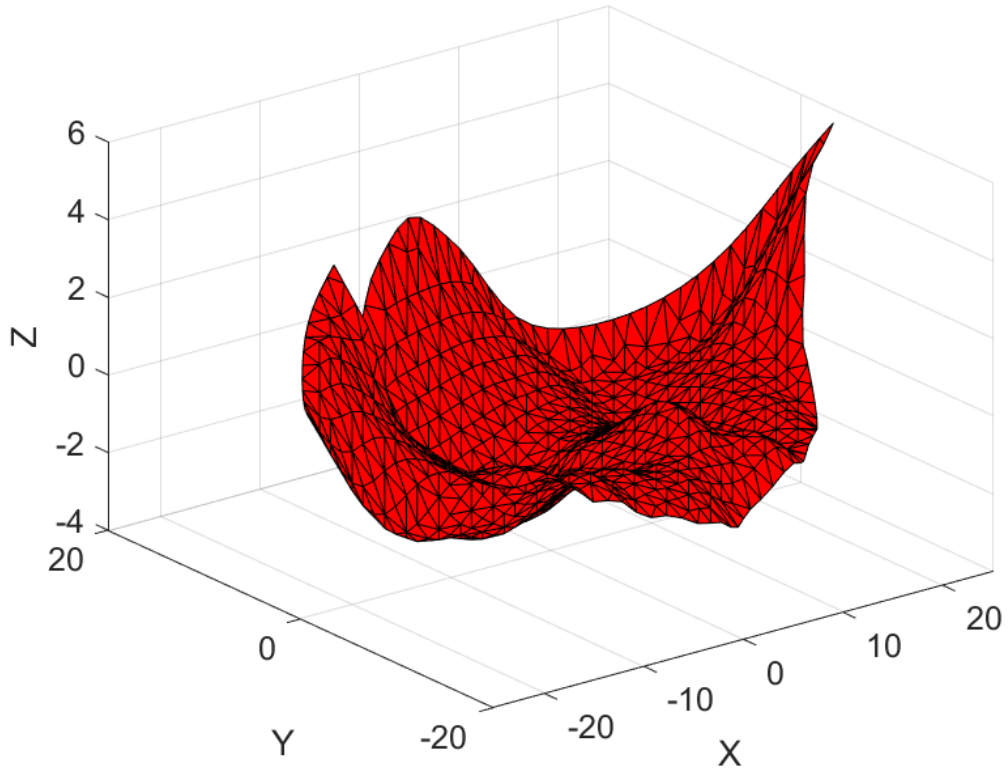


Figure 2.3: Anterior leaflet of new uniform data (891 points)

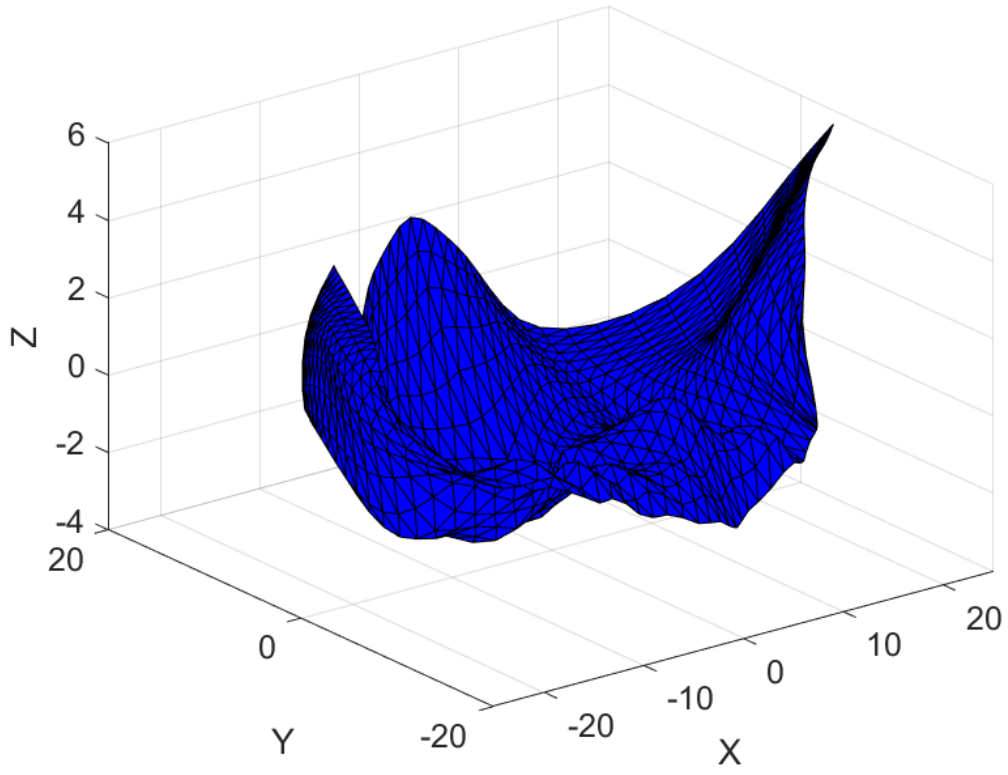


Figure 2.4: Anterior leaflet of original TOMTEC data (820 points)

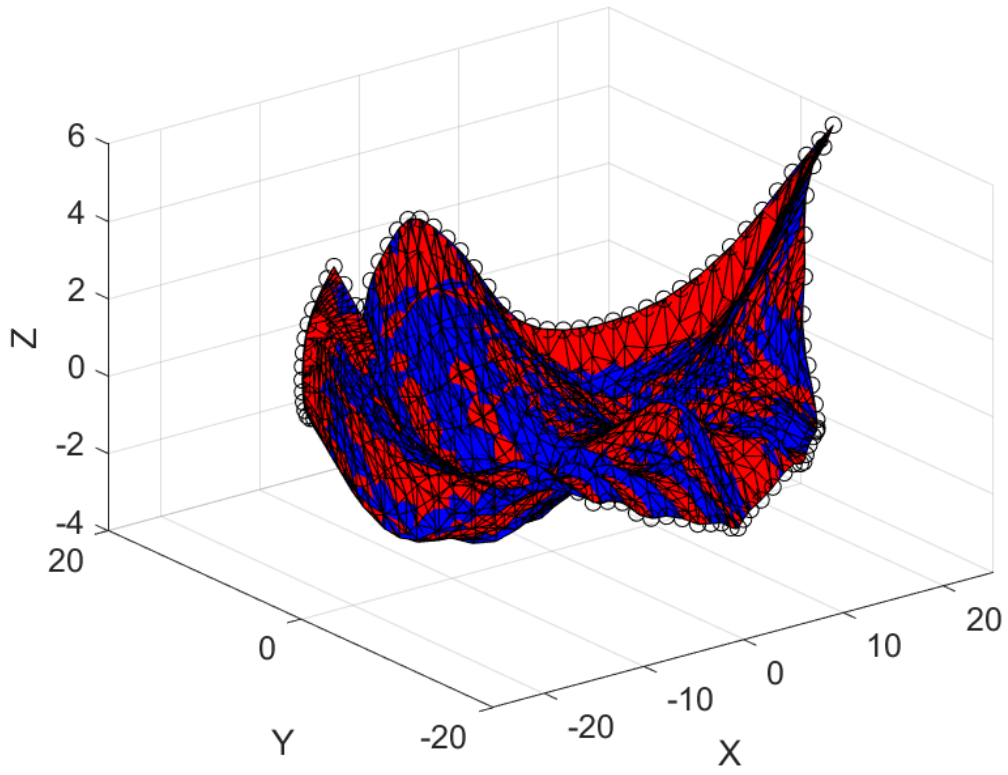


Figure 2.5: Comparison of uniform data and original TOMTEC data

2.5 Considering Total Frames

There are usually 5 to 18 frames for each echocardiograph during one heart beat. For pre-alignment, one can implement PCA on the first frame. Doing so we get the center and new coordinates, then apply this same change of coordinates to all other frames. After that, the prealignment is finished. This method works well in most cases. Another approach is to do PCA to each frame, then fit the leaflet boundary and surface by splines separately on each frame. After that, match separately the new coordinate system of each frame to the new coordinate system of the first frame. In practice, we prefer the first method. We require the next frame to have more points than the current frame, which means that the grid size N_r is an increasing sequence.

CHAPTER 3

Diffeomorphic Shape Matching

3.1 Dynamics of Diffeomorpic Flows

The continuous path defined by the time dependent R^3-diffeomorphisms F_t will be constructed as a geodesic $t \rightarrow F_t$ in a specific infinite dimensional Lie group of R^3-diffeomorphisms.

Fix a Hilbert space U of smooth vector fields on R^3 with norm $||\cdot||_U$; consider the associated Hilbert space $L^2(I, U)$ of vector field flows where $I \equiv [0, 1]$. The time dependent vector field flows $\nu : t \rightarrow \nu_t, 0 \leq t \leq 1$, where ν_t tends zero at infinity in

R^3 is associated with the flow equation

$$\frac{\partial F_t}{\partial t} = \nu_t(F_t), \quad t \in I, \tag{3.1}$$

$$F_0 = Id, \tag{3.2}$$

where Id is the identity map. G. Dupuis, U. Grenander, M. Miller, A. Trouvé, M. F. Beg, and L. Younes have presented this differential equation in papers [16, 40, 42, 8]. For some fixed constant $\lambda > 0$, the main variational problem is to find a vector field flow $\nu = (\nu_t)$ which minimizes the cost functional

$$J(\nu) = \int_0^1 ||\nu_t||_U^2 dt + \lambda \text{dis}[F_t(S_{ref}), S_{tar}], \tag{3.3}$$

where S_{ref} and S_{tar} are the reference surface and target surface. This objective linearly combines a kinetic energy term and a surface matching term. There are many known possible different Hilbert spaces and matching functionals, as shown by many deep research papers, like [20, 24, 19, 16].

This is a classical problem. There are three often used algorithms to compute large deformation diffeomorphisms. The main algorithms are various types of gradient descent algorithms including deterministic Newton descent. Christensen [10] has examined various boundary conditions, including zero, mixed and periodic situations. See also A. Trouve [42], Grenander and Miller [20]. A second type of algorithm is implemented by Christensen in [11, 35]. S. Joshi [32] implements landmark deformation in the Lagrangian framework.

In our research group (UH Mathematics), A. Jajoo has used the gradient descent with Armijo line search to minimize the cost functional (see [30]). Y. Qin has applied the second order Newton method to compute a cost minimizing diffeomorphic flow based on Bellman's optimal principle. She developed a dynamic programming algorithm, which works well when the grid size per surface snapshot is small [**?**]. For clinical application to mitral valves dynamics analysis in nearly real time, we had to implement computable algorithmics enabling very high speed and acceptable accuracy. We introduce here an operator splitting technique associated with Newton descent to minimize our cost functional. In brief, we break the optimization of our nonlinear control problem into two parts, an optimal quadratic control problem, which can be solved very efficiently by the Lagrangian method, and a set of single period optimization problems, which can be approximated in parallel by Newton iteration.

3.2 Existence and Uniqueness of Cost Minimizing Diffeomorphic Matching

The dynamic system (3.1) has a unique solution, when $t \to ||\nu_t||_U$ is integrable under suitable regularity conditions on the Hilbert space of U [16, 19].

Theorem 3.2.1 (Jajoo)**.** *Assume $\nu \in L^2(I, U)$ where U is continuously embedded in Sobolev space $W^{s,2}(R^3)$ for some $s > 5/2$, which is , then the dynamic system* (3.1),(3.2) *admits a unique solution F_t with each F_t being an R^3-diffeomorphism of smoothness class $1 \leq r \leq s - 3/2$.*

Proof. See [30] □

Thus we define the finite kinetic energy $Kin(\nu)$ as

$$Kin(\nu) = \frac{1}{2}||\nu||^2_{L^2(I,U)} = \frac{1}{2}\int_0^1 ||\nu_t||^2_U dt. \tag{3.4}$$

3.3 Reproducing Kernel Hilbert Spaces

In our context, the relevant Hilbert space U of R^3 vector fields is defined as a self-reproducing Hilbert space.

Definition 3.3.1. A function $K : R^3 \times R^3 \to R$ is called a semi-positive definite kernel iff

1. It is symmetric, i.e., $K(x, y) = K(y, x)$, for $x, y \in R^3$.
2. It is semi-positive definite, that is $\sum_{i=1}^{N}\sum_{j=1}^{N} c_i c_j K(x_i, x_j) \geq 0$ for an arbitrary positive integer N, any choice of N vectors $x_i \in R^3$, and any real values $c_i \in R$.

For any positive definite kernel K, we define a set of R^3-vector fields $w_{(z,u)}$ indexed by arbitrary pairs $(z, u) \in R^3 \times R^3$ as $w_{(z,u)}(x) = K(z, x)u$ for all $x \in R^3$, and we denote by LW the vector space of all finite linear combinations of the R^3-vector fields $w_{(z,u)}$. Then LW is endowed with the pre-Hilbertian scalar product

$$< w_{(z,u)}, w_{(z',u')} >= K(z, z') < u, u' >_{R^3} .$$

The reproducing kernel Hilbert space U classically defined by K is then the unique

Hilbert space generated by LW.

Definition 3.3.2. Let U be a Hilbert of functions defined on R^3. The real function $K(x, y)$ of x and y in R^3 is called a reproducing kernel for U if it satisfies:

1. For every y, $K_y(x) := K(x, y)$ is a function of x belonging to U.

2. The reproducing property: for every $y \in R^3$, and every $f \in U$, we have

$$f(y) = (f(x), K_y(x))_U.$$

There are many kernels to choose from, such as the Laplacian kernel, or the Gaussian kernel. Usually for shape matching applications, the radial Gaussian kernel K is often used; it is given by

$$K(x, y) = \exp(-\frac{||x - y||^2}{\sigma^2}),$$

with a suitable scale parameter $\sigma > 0$.

Definition 3.3.3. The Hilbert space H is said to be a Reproducing Kernel Hilbert Space, if there exists a reproducing kernel K for H.

Remark 3.3.4. By definition, choosing $x, y \in R^3$, we have

$$K_x(y) =< K_x, K_y >_H = K(y, x),$$
$$||K_x||_H = K(x, x)^{1/2}.$$

3.4 Discretization of the Flow Equation

Consider a time partition $0 = t_0 < t_1 < t_2 < \cdots < t_L = 1$, with step size $\tau_k = t_{k+1} - t_k$, $0 \leq k \leq L-1$. Assume that we are given $L+1$ snapshots, the shape snapshots $S_k = S(t_k)$. Represent each snapshot S_k by a finite grid of points $\{y_i(t_k) \in R^3 \mid 0 \leq i \leq M_k\}$. In our mitral valve application, the discretization of S_k will be computed by thin plate spline fitting as explained above. Deforming the initial snapshot S_0 by the R^3 diffeomorphism F_t^ν satisfying the flow equations (3.1), (3.2), we obtain at time t_k the surface $\hat{S}_k := F_{t_k}^\nu(S_0)$. Denote the trajectory of point $x_i(0)$ by $t \to x_i(t)$, $1 \leq i \leq N$. The discretization of $\hat{S}_k$ is $\{x_i(t_k) \in R^3 \mid 0 \leq i \leq N\}$ at time t_k. From (3.1) and (3.2), the ordinary differential equation for point $x_i(t)$ will be

$$\frac{dx_i(t)}{dt} = \nu_t(x_i(t)), \quad t \in (0,1], \tag{3.5}$$

here ν_t is the time dependent vector field of velocities.

The left side of equation (3.5) at t_k can be discretized by $x_i(t_{k+1})$ and $x_i(t_k)$ to give the approximation

$$\frac{dx_i(t)}{dt} \approx \frac{x_i(t_{k+1}) - x_i(t_k)}{t_{k+1} - t_k} \text{ at time } t_k.$$

On the right side of equation (3.5), a well known theorem in papers [9, 31] asserts that a cost minimizing vector field flow ν_t in U must be a linear combination of radial

Gaussian kernels

$$K(x, x') := \exp(-\frac{||x - x'||^2}{2\sigma^2}),$$

that is

$$\nu_t(x) = \sum_{n=1}^{N} K(x_n(t), x)\alpha_n(t), \quad x \in R^3,$$

here the time dependent vectors $\alpha_n(t) \in R^3$ are unknown coefficients. We will also specify the choice of σ later.

The equation (3.5) will be approximated by

$$\frac{x_i(t_{k+1}) - x_i(t_k)}{t_{k+1} - t_k} \approx \sum_{n=1}^{N} K(x_n(t_k), x_i(t_k))\alpha_n(t_k), \quad 0 \le i \le N, \;\; 0 \le k \le L-1, \tag{3.6}$$

which is

$$x_i(t_{k+1}) \approx x_i(t_k) + \tau_k \sum_{n=1}^{N} K(x_n(t_k), x_i(t_k))\alpha_n(t_k). \tag{3.7}$$

By the definition of a Reproducing Kernel Hilbert Space, we have the norm

$$||\nu_t(x)||^2 = \sum_{i=1}^{N}\sum_{j=1}^{N} K(x_i(t), x_j(t))\alpha_i(t)^T\alpha_j(t). \tag{3.8}$$

3.5 Distance between Shapes

To compare two surfaces S and $\hat{S}$, we need to define a distance between S and $\hat{S}$.

3.5.1 Hausdorff Distance

Define the distance from point x to surface S by,

$$\mathcal{D}(x, S) = \min_{y \in S} ||x - y||. \tag{3.9}$$

The classical Hausdorff disparities $h(S, \hat{S})$ and $h(\hat{S}, S)$ between two surfaces $S, \hat{S}$ are defined by

$$h(S, \hat{S}) = \max_{x \in S}(\mathcal{D}(x, \hat{S})),$$
$$h(\hat{S}, S) = \max_{x \in \hat{S}}(\mathcal{D}(S, x)).$$

The Hausdorff distance is $D_h(S, \hat{S}) = \max(h(S, \hat{S}), h(\hat{S}, S))$.

The shortcoming for the use of the distance D_h is that $D_h(S, \hat{S})$ is not a differentiable function of S and $\hat{S}$. But this distance is very useful to monitor the matching accuracy between a deformed initial surface and a target surface.

3.5.2 Dirac Measure

Consider a smooth compact surface S in R^3 discretized by a finite point grid x_i of large size N. The Lebesgue measure of R^3 induces on S a bounded measure which for large N can be well approximated by a linear combination μ of Dirac measures, as in [9, 31]:

$$\mu = \sum_{i=1}^{N} c_i \delta_{x_i},$$

here δ_{x_i} is the Dirac mass at node x_i, c_i is the unknown coefficients.

Consider any existing snapshot S_t and deformed surface $\hat{S}_t = F_t^\nu(S_0)$. Denote the corresponding measures by

$$\mu(S_0) = \sum_{i=1}^{N} a_i \delta_{x_i(0)}, \quad \mu(\hat{S}) = \sum_{i=1}^{N} a_i \delta_{x_i(t)}, \quad \mu(S) = \sum_{i=1}^{M} b_i \delta_{y_i(t)}, \tag{3.10}$$

here setting coefficients $a_i = \frac{1}{N}$, and $b_j = \frac{1}{M}$. Associated with another Gaussian kernel K_s for some positive suitable scale parameter s, the disparity function $\psi(F_t^\nu(S_0))$ between surface S_t and $\hat{S}_t$ is

$$\begin{aligned}
\psi(F_t^\nu(S_0)) &:= || \sum_{i=1}^{N} a_i \delta_{x_i(t)} - \sum_{j=1}^{M} b_j \delta_{y_j(t)} ||_H^2 \\
&=< \sum_{i=1}^{N} a_i \delta_{x_i(t)} - \sum_{j=1}^{M} b_j \delta_{y_j(t)}, \sum_{i=1}^{N} a_i \delta_{x_i(t)} - \sum_{j=1}^{M} b_j \delta_{y_j(t)} >_H \\
&=< \sum_{i=1}^{N} a_i \delta_{x_i(t)}, \sum_{i=1}^{N} a_i \delta_{x_i(t)} >_H - 2 < \sum_{i=1}^{N} a_i \delta_{x_i(t)}, \sum_{j=1}^{M} b_j \delta_{y_j(t)} >_H \\
&+ < \sum_{j=1}^{M} b_j \delta_{y_j(t)}, \sum_{j=1}^{M} b_j \delta_{y_j(t)} >_H \\
&= \sum_{i=1}^{N} \sum_{j=1}^{N} a_i a_j K_s(x_i(t), x_j(t)) - 2 \sum_{i=1}^{N} \sum_{j=1}^{M} a_i b_j K_s(x_i(t), y_j(t)) \\
&+ \sum_{i=1}^{M} \sum_{j=1}^{M} b_i b_j K_s(y_i(t), y_j(t)).
\end{aligned} \tag{3.11}$$

To quantify the matching accuracy of all discrete snapshots, the disparity cost functional is

$$\Psi(v) := \sum_{k=1}^{L} \lambda_k \psi(F_{t_k}^\nu(S_0)), \tag{3.12}$$

here λ_k is an adequately selected fixed positive weight. For the choice of λ_k, there are two special cases: Choice (1): uniformly deal with all snapshots by setting $\lambda_k = 1, 1 \leq k \leq L$. Choice (2): focus only on the end snapshot by setting $\lambda_L = 1$, and the other λ_k as 0.

3.6 The Choice of the Scale Parameters σ and s in Gaussian Kernels

Suppose S° is the finite grid discretizing the interior of the interior of an initial surface S. For each $p \in S^\circ$, and each fixed integer $r > 0$, define $\mathcal{N}(p) \subset S^\circ$ as the set of the r closest neighbors of p in S°, including p itself. The radius function is defined by

$$\mathcal{R}(p) = \max_{z \in \mathcal{N}(p)} ||p - z||, \forall p \in S^\circ.$$

Then the local mesh size of S° is

$$\mathcal{M}(S^\circ) = \max_{p \in S^\circ} \{\mathcal{R}(p)\}.$$

The scale parameter σ for the radial Gaussian kernel will be selected as follows

$$\sigma = \frac{coef_1 \times \mathcal{M}(S^\circ)}{\sqrt{2}},$$

where $coef_1$ is some constant such that $1 \leq coef_1 \leq 2$.

For selecting the s value, we use the Hausdorff distance between $\hat{S}$ and S

$$s = coef_2 \times \frac{D_h(\hat{S}, S)}{\sqrt{2}},$$

here $1 \leq coef_2 \leq 2$.

In our experiments, s is a variable depending on how close the deformed surface is to the target surface.

In practice, we introduce high quantiles instead of absolute maxima to avoid possible strange values around the boundary. For instance, $\mathcal{M}(S^\circ)$ will be

$$\mathcal{M}(S^\circ) := 80\text{th percentile of } \{\mathcal{R}(p) | p \in S^\circ\}.$$

Similarly, $D_h(\hat{S}, S)$ will be changed to

$$D_h(\hat{S}, S) := 98\text{th percentile of } \{\mathcal{D}(x, S) | x \in \hat{S}\}.$$

3.7 Existence of a Solution for the Cost Minimization Problem

Using the kinetic energy $Kin(\nu)$ defined in (3.4) and the disparity function $\psi(F_t^\nu(S_0))$ defined in (3.11), the cost function $J : L^2(I, U) \to R$ introduced in papers [1, 22] is

$$J(\nu) := \lambda_0 Kin(\nu) + \int_0^1 \psi(F_t^\nu(S_0))dt, \tag{3.13}$$

here λ_0 is a fixed weight to be determined later on.

Theorem 3.7.1. *The minimization problem*

$$\inf_{\nu \in L^2(I,U)} J(\nu),$$

under the non-linear constraints specified by the ODE system (3.1),(3.2) *has a solution.*

Proof. See [2]. □

CHAPTER 4

Optimal Nonlinear Control Problem

4.1 Problem Description

The notations are

$$\boldsymbol{x_k} := \boldsymbol{x(t_k)} = (x_1(t_k), \cdots, x_N(t_k)) \in R^{3N},$$

$$\boldsymbol{y_k} := (y_1(t_k), \cdots, y_M(t_k)) \in R^{3M} \quad \text{is target data,}$$

$$\boldsymbol{\alpha_k} := \boldsymbol{\alpha(t_k)} = (\alpha_1(t_k), \cdots, \alpha_M(t_k)) \in R^{3M},$$

$$\boldsymbol{x} := (\boldsymbol{x_0}, \boldsymbol{x_1}, \cdots, \boldsymbol{x_L}) \in R^{3N(L+1)},$$
$$\boldsymbol{y} := (\boldsymbol{y_0}, \boldsymbol{y_1}, \cdots, \boldsymbol{y_L}) \in R^{3M(L+1)},$$
$$\boldsymbol{\alpha} := (\boldsymbol{\alpha_0}, \boldsymbol{\alpha_1}, \cdots, \boldsymbol{\alpha_L}) \in R^{3M(L+1)},$$

$\mathbb{K}(\boldsymbol{x_k}) = [\mathcal{K}_{ij}(x(t_k))] \in R^{3N\times 3M}$ the $N \times M$ block matrix of $\mathcal{K}_{ij}(x(t_k))$,

$\mathcal{K}_{ij}(x(t_k)) = K_s(x_i(t_k), y_j(t_k))I_3 \in R^{3\times 3}$,

$\mathbb{K}(\boldsymbol{y_k}) = [\mathcal{K}_{ij}(y(t_k))] \in R^{3M\times 3M}$ the $M \times M$ block matrix of $\mathcal{K}_{ij}(y(t_k))$,

$\mathcal{K}_{ij}(y(t_k)) = K(y_i(t_k), y_j(t_k))I_3 \in R^{3\times 3}$,

$0 \le k \le L$.

The purpose is to minimize the discrete cost functional:

$$\text{minimize} \quad J(\{\boldsymbol{\alpha_k}\}_{k=0}^{L}; x_{init}) := \lambda_0 \sum_{k=0}^{L-1} \phi(\boldsymbol{x_k}, \boldsymbol{\alpha_k}; t_k) + \sum_{k=1}^{L} \lambda_k \psi(\boldsymbol{x_k}), \tag{4.1}$$

$$\text{subject to} \quad \boldsymbol{x_{k+1}} = \boldsymbol{x_k} + \tau_k \mathbb{K}(\boldsymbol{x_k})\boldsymbol{\alpha_k}, \quad k = 0, \cdots, L-1, \tag{4.2}$$

$$\boldsymbol{x_0} = \boldsymbol{x}_{init}.$$

The kinetic energy and the snapshots matching terms are given by

$$\phi(\boldsymbol{x_k}, \boldsymbol{\alpha_k}; t_k) = \frac{\tau_k}{2} \boldsymbol{\alpha_k}^T \mathbb{K}(\boldsymbol{y_k}) \boldsymbol{\alpha_k}. \tag{4.3}$$

$$\begin{aligned} \psi(\boldsymbol{x_k}) = \frac{1}{N^2} \sum_{i,j=1}^{N} K_s(x_i(t_k), x_j(t_k)) - \frac{2}{NM} \sum_{i=1}^{N} \sum_{j=1}^{M} K_s(x_i(t_k), y_j(t_k)) \\ + \frac{1}{M^2} \sum_{i,j=1}^{M} K_s(y_i(t_k), y_j(t_k)). \end{aligned} \tag{4.4}$$

Combining them, we get the discrete cost functional in detail

$$\begin{aligned} J(\{\boldsymbol{\alpha_k}\}_{k=1}^{L}) = & \sum_{k=0}^{L-1} \frac{\tau_k}{2} \boldsymbol{\alpha_k}^T \mathbb{K}(\boldsymbol{y_k}) \boldsymbol{\alpha_k} + \sum_{k=1}^{L} \lambda_k \Big\{ \frac{1}{N^2} \sum_{i,j=1}^{N} K_s(x_i(t_k), x_j(t_k)) \\ & - \frac{2}{NM} \sum_{i=1}^{N} \sum_{j=1}^{M} K_s(x_i(t_k), y_j(t_k)) + \frac{1}{M^2} \sum_{i,j=1}^{M} K_s(y_i(t_k), y_j(t_k)) \Big\}. \end{aligned}$$

4.2 Splitting Technique

For convenience, setting $\Phi(\boldsymbol{x}, \boldsymbol{\alpha}) := \sum_{k=0}^{L-1} \phi(\boldsymbol{x_k}, \boldsymbol{\alpha_k}; t_k)$ and $\Psi(\boldsymbol{x}, \boldsymbol{\alpha}) = \sum_{k=1}^{L} \lambda_k \psi(\boldsymbol{x_k})$, the cost function to minimize then becomes

$$J(\boldsymbol{x}, \boldsymbol{\alpha}) = \Phi(\boldsymbol{x}, \boldsymbol{\alpha}) + \Psi(\boldsymbol{x}, \boldsymbol{\alpha}).$$

We try to decouple the minimizations of Φ and Ψ as follows

$$\begin{aligned} &\text{minimize} \quad J(\boldsymbol{x}, \boldsymbol{\alpha}) = \Phi(\boldsymbol{x}, \boldsymbol{\alpha}) + \Psi(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{\alpha}}), \\ &\text{subject to} \quad ||(\boldsymbol{x}, \boldsymbol{\alpha}) - (\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{\alpha}})|| < \epsilon, \\ &\qquad\qquad \boldsymbol{x_{k+1}} = \boldsymbol{x_k} + \tau_k \mathbb{K}(\boldsymbol{x_k})\boldsymbol{\alpha_k}, \quad k = 0, \cdots, L-1. \end{aligned}$$

Definition 4.2.1. The proximal operator $\boldsymbol{prox}_f(v) : R^d \to R^d$ of f is defined by

$$\boldsymbol{prox}_f(v) = \arg\min_x (f(x) + (1/2)||x - v||^2). \tag{4.5}$$

The proximal minimization algorithm, also called proximal iteration, is $x^{n+1} := \boldsymbol{prox}_f(x^n)$ where $f : R^d \to R \cup \{+\infty\}$ is a closed proper convex function, n is the iteration counter, and x^n denotes the nth iterate of the algorithm. If f has a minimum, then x^n will converge to the minimizer of the function f and $f(x^n)$ converges to the minimum of f (see [4]).

The Alternating Direction Method of Multipliers (ADMM), also known as Douglas-Rachford splitting, starts from any $(\tilde{\boldsymbol{x}}^0, \tilde{\boldsymbol{\alpha}}^0)$, (z^0, β^0) and implements the following iterative steps for $n = 0, 1, \cdots$

$$\begin{aligned} (\boldsymbol{x}^{n+1}, \boldsymbol{\alpha}^{n+1}) &:= \boldsymbol{prox}_{\rho\Phi + I_{\mathcal{D}}}((\tilde{\boldsymbol{x}}^n, \tilde{\boldsymbol{\alpha}}^n) + (z^n, \beta^n)), \\ (\tilde{\boldsymbol{x}}^{n+1}, \tilde{\boldsymbol{\alpha}}^{n+1}) &:= \boldsymbol{prox}_{\rho\Psi}((\boldsymbol{x}^{n+1}, \boldsymbol{\alpha}^{n+1}) - (z^n, \beta^n)), \\ (z^{n+1}, \beta^{n+1}) &:= (z^n, \beta^n) + (\tilde{\boldsymbol{x}}^{n+1}, \tilde{\boldsymbol{\alpha}}^{n+1}) - (\boldsymbol{x}^{n+1}, \boldsymbol{\alpha}^{n+1}), \end{aligned}$$

where $\mathcal{D} = \{(\boldsymbol{x}, \boldsymbol{\alpha}) | \boldsymbol{x_0} = \boldsymbol{x}_{init}, \quad \boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \tau_k \mathbb{K}(\boldsymbol{x_k})\boldsymbol{\alpha_k}, \quad 0 \leq k \leq L-1\}$ is the set of state-control pairs that satisfies the dynamics, $\rho > 0$ is the parameter. (z, β) is a pair of dual variables associated with the consensus constraint $||(\boldsymbol{x}, \boldsymbol{\alpha}) - (\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{\alpha}})|| < \epsilon$, (z^{n+1}, β^{n+1}) is the running sum of errors $(\boldsymbol{x}, \boldsymbol{\alpha}) - (\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{\alpha}})$ and converges to zero. In other words, $(\boldsymbol{x}^n, \boldsymbol{\alpha}^n) - (\tilde{\boldsymbol{x}}^n, \tilde{\boldsymbol{\alpha}}^n)$ tends to 0 as n tends to infinity. This technique converges under very general conditions [7]. The advantage of ADMM is that the objective terms are handled completely separately, and indeed, the functions are accessed only through their proximal operators. ADMM is most useful when the proximal operators of Φ and Ψ can be efficiently evaluated while the proximal operator for the sum of Φ and Ψ is not easy to evaluate.

The optimal control problem (4.1), (4.2) will hence be solved here by the following explicit version of the Douglas-Rachford Splitting Method:

$$(\boldsymbol{x}^{n+1}, \boldsymbol{\alpha}^{n+1}) = \underset{(\boldsymbol{x}, \boldsymbol{\alpha}) \in \mathcal{D}}{\arg\min} (\Phi(\boldsymbol{x}, \boldsymbol{\alpha}) + \frac{\rho}{2} ||(\boldsymbol{x}, \boldsymbol{\alpha}) - (\tilde{\boldsymbol{x}}^n, \tilde{\boldsymbol{\alpha}}^n) - (z^n, \beta^n)||_2^2), \tag{4.6}$$

$$(\tilde{\boldsymbol{x}}^{n+1}, \tilde{\boldsymbol{\alpha}}^{n+1}) = \underset{(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{\alpha}})}{\arg\min} (\Psi(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{\alpha}}) + \frac{\rho}{2} ||(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{\alpha}}) - (\boldsymbol{x}^{n+1}, \boldsymbol{\alpha}^{n+1}) + (z^n, \beta^n)||_2^2), \tag{4.7}$$

$$(z^{n+1}, \beta^{n+1}) = (z^n, \beta^n) + (\tilde{\boldsymbol{x}}^{n+1}, \tilde{\boldsymbol{\alpha}}^{n+1}) - (\boldsymbol{x}^{n+1}, \boldsymbol{\alpha}^{n+1}), \tag{4.8}$$

with variables $(\boldsymbol{x}, \boldsymbol{\alpha})$, $(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{\alpha}}) \in R^{3N(L+1)+3M(L+1)}$, and $\rho > 0$ is an algorithm parameter. This formalization is referred as consensus form in papers [15, 7, 23].

$$
\begin{aligned}
\Phi(\boldsymbol{x},\boldsymbol{\alpha}) :&= \sum_{k=0}^{L-1} \phi(\boldsymbol{x_k},\boldsymbol{\alpha_k};t_k) \\
&= \lambda_0 \sum_{k=0}^{L-1} \frac{\tau_k}{2} \boldsymbol{\alpha_k}^T \mathbb{K}(\boldsymbol{y_k}) \boldsymbol{\alpha_k} \\
&= \lambda_0 \sum_{k=0}^{L-1} \frac{\tau_k}{2} \begin{bmatrix} \boldsymbol{x_k} \\ \boldsymbol{\alpha_k} \end{bmatrix}^T \begin{bmatrix} 0 & 0 \\ 0 & \mathbb{K}(\boldsymbol{y_k}) \end{bmatrix} \begin{bmatrix} \boldsymbol{x_k} \\ \boldsymbol{\alpha_k} \end{bmatrix} \\
&= \frac{\lambda_0}{2} \begin{bmatrix} \boldsymbol{x_0} \\ \boldsymbol{\alpha_0} \\ \vdots \\ \boldsymbol{x_{L-1}} \\ \boldsymbol{\alpha_{L-1}} \\ \boldsymbol{x_L} \\ \boldsymbol{\alpha_L} \end{bmatrix}^T \begin{bmatrix} 0 & 0 & \cdots & 0 & 0 & 0 & 0 \\ 0 & \tau_0\mathbb{K}(\boldsymbol{y_0}) & \cdots & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & 0 & 0 & 0 \\ 0 & 0 & \cdots & 0 & \tau_{L-1}\mathbb{K}(\boldsymbol{y_{L-1}}) & 0 & 0 \\ 0 & 0 & \cdots & 0 & 0 & 0 & 0 \\ 0 & 0 & \cdots & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{x_0} \\ \boldsymbol{\alpha_0} \\ \vdots \\ \boldsymbol{x_{L-1}} \\ \boldsymbol{\alpha_{L-1}} \\ \boldsymbol{x_L} \\ \boldsymbol{\alpha_L} \end{bmatrix}.
\end{aligned}
$$

4.3 Quadratic Control Step

The first step (4.6) in the splitting algorithm

$$
\underset{(\boldsymbol{x},\boldsymbol{\alpha})\in\mathcal{D}}{\text{minimize}} (\Phi(\boldsymbol{x},\boldsymbol{\alpha}) + \frac{\rho}{2} ||(\boldsymbol{x},\boldsymbol{\alpha}) - (\tilde{\boldsymbol{x}}^n,\tilde{\boldsymbol{\alpha}}^n) - (z^n,\beta^n)||^2),
$$

can be expressed in matrix form and the objective function is a quadratic function:

$$\begin{aligned} \text{minimize} \quad & \frac{1}{2}\omega^T E\omega + \eta^T\omega, \\ \text{subject to} \quad & G\omega = h. \end{aligned} \tag{4.9}$$

Here the vectors $\omega, \eta \in R^{(3N+3M)(L+1)}$, $h \in R^{3N(L+1)}$, and the matrices $E \in R^{(3N+3M)(L+1)\times(3N+3M)(L+1)}$, $G \in R^{3N(L+1)\times(3N+3M)(L+1)}$ are given by

$$\omega = \begin{bmatrix} \boldsymbol{x_0} \\ \boldsymbol{\alpha_0} \\ \vdots \\ \boldsymbol{x_L} \\ \boldsymbol{\alpha_L} \end{bmatrix}, \quad \eta = \begin{bmatrix} -\rho(\tilde{\boldsymbol{x}}_0^n + z_0^n) \\ -\rho(\tilde{\boldsymbol{\alpha}}_0^n + \beta_0^n) \\ \vdots \\ -\rho(\tilde{\boldsymbol{x}}_L^n + z_L^n) \\ -\rho(\tilde{\boldsymbol{\alpha}}_L^n + \beta_L^n) \end{bmatrix}, \quad h = \begin{bmatrix} \boldsymbol{x}_{init} \\ 0 \\ \vdots \\ 0 \end{bmatrix},$$

$$E = \begin{bmatrix} \rho I & 0 & \cdots & 0 & 0 \\ 0 & \lambda\tau_0\mathbb{K}(\boldsymbol{y_0}) + \rho I & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & \rho I & 0 \\ 0 & 0 & \cdots & 0 & \rho I \end{bmatrix},$$

and

$$G = \begin{bmatrix} I & 0 & 0 & \cdots & 0 & 0 & 0 & 0 \\ -I & -\tau_0 \mathbb{K}(\boldsymbol{y_0}) & I & \cdots & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & I & 0 & 0 & 0 \\ 0 & 0 & 0 & \cdots & -I & -\tau_{L-1} \mathbb{K}(\boldsymbol{y_{L-1}}) & I & 0 \end{bmatrix}.$$

In order to solve this quadratic optimization problem, we use the method of Lagrange multipliers. Define the Lagrange function

$$\mathcal{L}(\omega, \Lambda) = \frac{1}{2}\omega^T E\omega + \eta^T \omega + \Lambda^T (G\omega - h).$$

The KKT conditions for the solution ω^* and Λ^* of this quadratic programming problem give rise to the following linear system

$$\begin{bmatrix} E & G^T \\ G & 0 \end{bmatrix} \begin{bmatrix} \omega^* \\ \Lambda^* \end{bmatrix} = \begin{bmatrix} -\eta \\ h \end{bmatrix}.$$

To make sure the matrix is invertible, we will add a small ϵ on the 0 block in the KKT matrix. Suppose we can apply the LDL^T decomposition to the regularized

KKT,

$$\begin{bmatrix} E & G^T \\ G & \epsilon \end{bmatrix} = PLDL^T P^T. \tag{4.10}$$

Then the solution could be expressed as

$$\begin{bmatrix} \omega^* \\ \Lambda^* \end{bmatrix} = PL^{-T}D^{-1}L^{-1}P^T \begin{bmatrix} -\eta \\ h \end{bmatrix}. \tag{4.11}$$

Then we can get $(\boldsymbol{x}^{n+1}, \boldsymbol{\alpha}^{n+1})$ from ω^*.

In practice, we initialize $\tilde{\boldsymbol{x}}^1, \tilde{\boldsymbol{\alpha}}^1, z^1$ and β^1 as follows: $\tilde{\boldsymbol{x}}^1 = (\boldsymbol{x}_0, \boldsymbol{x}_0, \cdots, \boldsymbol{x}_0)$, $\tilde{\boldsymbol{\alpha}}^1 = \mathbf{0}$, $z^1 = \mathbf{0}$, $\beta^1 = \mathbf{0}$.

4.4 Single Period Proximal Step

In (4.7), the second minimization problem of the splitting algorithm is

$$\underset{(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{\alpha}})}{\text{minimize}} (\Psi(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{\alpha}}) + \frac{\rho}{2} ||(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{\alpha}}) - (\boldsymbol{x}^{n+1}, \boldsymbol{\alpha}^{n+1}) + (z^n, \beta^n)||^2),$$

where

$$\Psi(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{\alpha}}) = \sum_{k=1}^{L} \lambda_k \psi(\tilde{\boldsymbol{x}}_{\boldsymbol{k}}).$$

Here, $\psi(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{\alpha}})$ is separable across state and control. It can be split further into L proximal optimization problems. So (4.7) can be changed to find the minimizing

value of $\tilde{\boldsymbol{x}}_{\boldsymbol{k}}$, $\tilde{\boldsymbol{\alpha}}_{\boldsymbol{k}}$ as follows:

$$\underset{(\tilde{\boldsymbol{x}}_k, \tilde{\boldsymbol{\alpha}}_k)}{\text{minimize}} \quad \lambda_k \psi(\tilde{\boldsymbol{x}}_k) + \frac{\rho}{2} ||(\tilde{\boldsymbol{x}}_k, \tilde{\boldsymbol{\alpha}}_{\boldsymbol{k}}) - (\boldsymbol{x}_k^{n+1} - z_k^n, \boldsymbol{\alpha}_k^{n+1} - \beta_k^n)||_2^2. \tag{4.12}$$

Hence, the minimizing value for $\tilde{\boldsymbol{\alpha}}_{\boldsymbol{k}}$ has the form $\tilde{\boldsymbol{\alpha}}_{\boldsymbol{k}} = \boldsymbol{\alpha}_k^{n+1} - \beta_k^n$. To find the minimizing value for $\tilde{\boldsymbol{x}}_k$, we have to solve the following optimization problem:

$$\underset{\tilde{\boldsymbol{x}}_k}{\text{minimize}} \quad \text{SP}_k(\tilde{\boldsymbol{x}}_k) := \lambda_k \psi(\tilde{\boldsymbol{x}}_k) + \frac{\rho}{2} (\tilde{\boldsymbol{x}}_k - (\boldsymbol{x}_k^{n+1} - z_k^n))^2. \tag{4.13}$$

Simplify the notation as follows:

$$\text{SP}_k(\tilde{\boldsymbol{x}}_k) = \lambda_k \psi(\tilde{\boldsymbol{x}}_k) + \frac{\rho}{2} (\tilde{\boldsymbol{x}}_k - w_k)^2, \tag{4.14}$$

where $w_k = \boldsymbol{x}_k^{n+1} - z_k^n$. Our goal is to find a local minimizer for $\tilde{\boldsymbol{x}}_k$.

We use Newton's method [37], which is displayed in Algorithm 1 to compute the minimal value of the function $\text{SP}_k(\tilde{\boldsymbol{x}}_k)$.

Algorithm 1 Pure Newton's method

1: Select and fix a tolerance parameter $\epsilon^{newton} > 0$;
Require: : initialize $\xi^0 \in R^{2N}$ arbitrarily.
2: $l = 1$.
3: compute the Newton direction d_l, which satisfies $\nabla^2 \text{SP}_k(\xi^l) d_l = -\nabla \text{SP}_k(\xi^l)$;
4: set $\xi^{l+1} = \xi^l + d_l$;
5: **if** $||\nabla \text{SP}_k(\xi^{l+1})|| < \epsilon^{newton}$ **then**
6: stop and ξ^{l+1} will be the local minimum assigned to the output $\tilde{\boldsymbol{x}}_k^{n+1}$
7: **else**
8: return to step 2, and set $l = l + 1$;

We can run Newton's method L times to get $(\tilde{\boldsymbol{x}}^{n+1}, \tilde{\boldsymbol{\alpha}}^{n+1})$ in (4.7).

The first and second derivatives of SP_k are needed in the preceding Newton descent,

$$[(\mathrm{SP}_k)_x]_u = \frac{2\lambda_k}{N^2}\sum_{j=1}^{N} K_s(x^u, x^j)(-\frac{x^u - x^j}{s^2}) - \frac{2\lambda_k}{NM}\sum_{j=1}^{M} K_s(x^u, y^j)(-\frac{x^u - y^j}{s^2}) + \rho(x^u - w^u),$$

and

$$[(\mathrm{SP}_k)_{xx}]_{uv} = \delta_u^v[\frac{2\lambda_k}{N^2}\sum_{j=1}^{N} K_{11}(x^u, x^j) - \frac{2\lambda_k}{NM}\sum_{j=1}^{M} K_{11}(x^u, y^j) + \rho I_{3\times 3}] + \frac{2\lambda_k}{N^2}K_{12}(x^u, x^v),$$

where

$$\begin{aligned} K_{11}(x^u, x^j) &= \frac{\partial}{\partial x^u}[-K_s(x^u, x^j)\frac{(x^u - x^j)}{s^2}] \\ &= K_s(x^u, x^j)\frac{(x^u - x^j)'(x^u - x^j)}{s^4} - K_s(x^u, x^j)\frac{I_{3\times 3}}{s^2}, \end{aligned}$$

$$\begin{aligned} K_{11}(x^u, y^j) &= \frac{\partial}{\partial x^u}[-K_s(x^u, y^j)\frac{(x^u - y^j)}{s^2}] \\ &= K_s(x^u, y^j)\frac{(x^u - y^j)'(x^u - y^j)}{s^4} - K_s(x^u, y^j)\frac{I_{3\times 3}}{s^2}, \end{aligned}$$

and

$$K_{12}(x^u, x^v) = \frac{\partial}{\partial x^v}[-K_s(x^u, x^v)\frac{(x^u - x^v)}{s^2}] \\ = -K_s(x^u, x^v)\frac{(x^u - x^v)'(x^u - x^v)}{s^4} + K_s(x^u, x^v)\frac{I_{3\times 3}}{s^2}.$$

4.5 Relaxation and Regularization

To improve the convergence rate of the preceding operator splitting technique, we apply a relaxation approach, which replaces $(\boldsymbol{x}^{n+1}, \alpha^{n+1})$ in (4.7) and (4.8) by

$$(\boldsymbol{x}_{new}^{n+1}, \boldsymbol{\alpha}_{new}^{n+1}) = a(\boldsymbol{x}^{n+1}, \boldsymbol{\alpha}^{n+1}) + (1-a)(\tilde{\boldsymbol{x}}^n, \tilde{\boldsymbol{\alpha}}^n), \tag{4.15}$$

where $a \in (0, 1)$ is a relaxation parameter.

To ensure the factorization always exists and the factorization algorithm is stable, we need to regularize the system. Instead of the original KKT matrix, we factor the regularized KKT matrix as indicated in (4.10).

4.6 Convergence and Stopping Criteria

The primal residual and dual residual for (4.1),(4.2) are defined by

$$re^n = (\boldsymbol{x}^n, \boldsymbol{\alpha}^n) - (\tilde{\boldsymbol{x}}^n, \tilde{\boldsymbol{\alpha}}^n), \tag{4.16}$$

$$dre^n = \rho((\tilde{\boldsymbol{x}}^n, \tilde{\boldsymbol{\alpha}}^n) - (\tilde{\boldsymbol{x}}^{n-1}, \tilde{\boldsymbol{\alpha}}^{n-1})). \tag{4.17}$$

Our stopping criterion is to stop iterations when the residuals are smaller than some thresholds

$$||re^n||_2 \leq \epsilon^{pri}, \quad ||dre^n||_2 \leq \epsilon^{dual},$$

where $\epsilon^{pri} > 0, \epsilon^{dual} > 0$ are tolerances for primal and dual feasibility. The actual values of those two tolerances will be set as follows

$$\epsilon^{pri} = \epsilon^{abs}\sqrt{(L+1)(N+M)} + \epsilon^{rel}\max\{||(\boldsymbol{x}^n, u^n)||_2, ||(\tilde{\boldsymbol{x}}^n, \tilde{\boldsymbol{\alpha}}^n)||_2\}, \tag{4.18}$$

$$\epsilon^{dual} = \epsilon^{abs}\sqrt{(L+1)(N+M)} + \epsilon^{rel}||(z^n, \beta^n)||_2, \tag{4.19}$$

where $\epsilon^{abs} > 0$ and $\epsilon^{rel} > 0$ are absolute and relative tolerances respectively.

CHAPTER 5

Application to Concrete Diffeomorphic Shape Matching

Examples

All computations are carried out on a system of dual core Intel i7-6600u with CPU 2.6GHz and 2.81GHz, and 8GB of RAM, running Windows 10. All our codes are implemented in Matlab.

5.1 Matching from Plane to Sphere

The initial and target surfaces y_0 and y_1 are respectively a planar disc of radius 1 and an upper hemisphere of radius 2 in R^3. For simplicity, the time partition is $t_0 = 0$ and $t_1 = 1$. A spherical coordinate system is used to generate a point grid of size n_y on the upper hemisphere of radius 2 in R^3. Fix an integer n_y. Define two vectors of n_y angles, namely $\theta_y = [0, \frac{\pi}{2(n_y-1)}, \cdots, \frac{\pi}{2(n_y-1)}, \frac{\pi(n_y-2)}{2(ny-1)}] \in R^{(n_y-1)}$, and $\varphi_y = [0, \frac{2\pi}{n_y}, \cdots, 2\pi] \in R^{n_y}$. The initial planar disc of radius $r_y(t_0) = 1$ in R^3 is discretized by the following set of $n_y(n_y - 1)$ points

$$y_{(i,j)}(t_0) = [r_y(t_0)\cos(\theta_y(i))cos(\varphi_y(j)), r_y(t_0)\cos(\theta_y(i))\sin(\varphi_y(j)), 0],$$

$1 \leq i \leq n_y - 1$, $1 \leq j \leq n_y$. Define similarly two other vectors of n_x angles denoted by θ_x and φ_x, fix $r_x(t_0) = 1$ and define another finite planar grid of points by

$$x_{(i,j)}(t_0) = [r_x(t_0)\cos(\theta_x(i))cos(\varphi_x(j)), r_x(t_0)\cos(\theta_x(i))\sin(\varphi_x(j)), 0],$$

$1 \leq i \leq n_x - 1$, $1 \leq j \leq n_x$.

The discretized grid on the upper hemisphere is given by

$$y_{(i,j)}(t_1) = [r_y(t_1)\cos(\theta_y(i))cos(\varphi_y(j)), r_y(t_1)\cos(\theta_y(i))\sin(\varphi_y(j)), \quad r_y(t_1)\sin(\theta_y(i))].$$

where $1 \leq i \leq n_y - 1$, $1 \leq j \leq n_y$.

These point grids are represented in matrix form as $\boldsymbol{y_k} = (y_1(t_k), \cdots, y_{N_y}(t_k))$,

$\boldsymbol{x_0} = (x_1(t_0), \cdots, x_{N_x}(t_0))$. The radiuses are fixed as follows: $r_y(t_0) = 1$, $r_y(t_1) = 2$ and $r_x(t_0) = 1$.

Starting from the initial $\boldsymbol{x_0}$, and applying the operator splitting method, we get the approximation $\boldsymbol{x_1}$ and compare it to $\boldsymbol{y_1}$. Table 5.1 shows the three simulation results based on three different grid sizes for the target surface; the corresponding three approximating surfaces $\boldsymbol{x_1}$ are given in Figure 5.1. The first conclusion is that these approximations are all close to the targeted upper hemisphere with radius 2. The second conclusion is that computing times remain fairly stable when one increases the number of points in the target surface.

Table 5.1: Operator splitting method for diffeomorphic matching of a planar disc onto a hemisphere

Number of grid points for $\boldsymbol{x_0}$	182	182	182
Number of grid points for $\boldsymbol{y_0}$	380	462	600
Radius of hemisphere	2	2	2
Horizon length L	2	2	2
Fixed number of iterations	200	200	200
KKT solving time (ms)	0.38	0.35	1.69
Quadratic control step time (ms)	0.013	0.0158	0.0642
Single period proximal step time (ms)	0.132	0.157	0.324
Shape matching disparity	4×10^{-6}	4×10^{-6}	8×10^{-6}
Number of Newton iteration	4	4	4

We put the three approximations of $\boldsymbol{x_1}$ with different mesh sizes of the target upper hemisphere $\boldsymbol{y_1}$ in the same Figure 5.1. They are very close to each other.

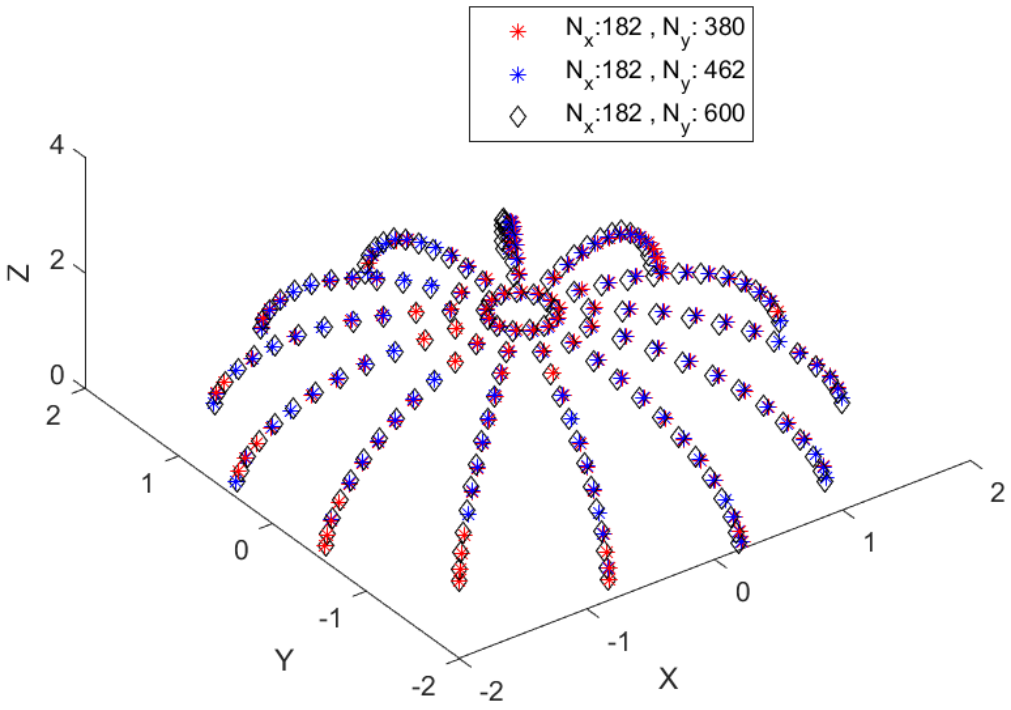


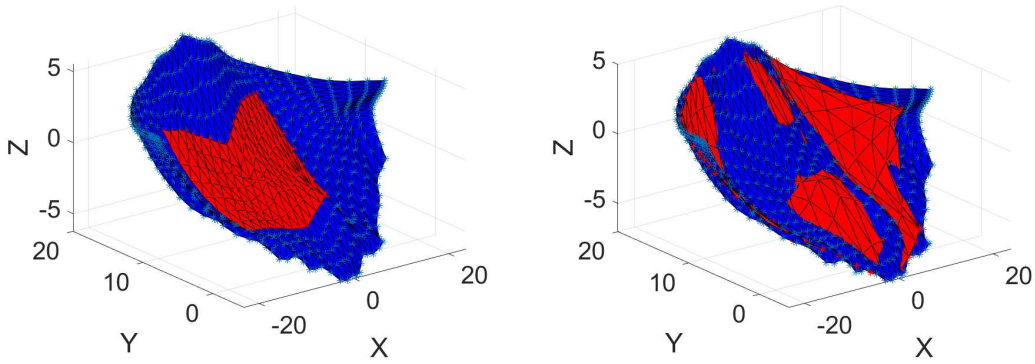
Figure 5.1: Diffeomorhic matching of a planar disc onto an upper hemisphere

5.2 Diffeomorphic Matching from Surface to Surface

Consider the sequence of discretized surface data extracted from echocardiographic 3D movies of mitral valve dynamics. We consider here the mitral valve data corresponding to a single cardiology patient. We have a sequence of 6 discretized mitral valve snapshots $\boldsymbol{y_k}, 0 \leq k \leq 5$ extracted (via TOMTEC softwares) by Carlos El Tallawi at Houston Methodist Hospital (Cardiology). A divided part of $\boldsymbol{y_0}$ is defined as initial data $\boldsymbol{x_0}$. We reconstruct an associated diffeomorphic flow matching these six snapshots by the operator splitting technique described in preceding chapters. This generates reconstructed trajectories for each initial grid point in $\boldsymbol{x_k}$. Table 5.2 shows two approximation results corresponding to different grid sizes for the discretization of the initial surface $\boldsymbol{x}_0$. We conclude from Figures 5.2 and 5.3 that for each one of the snapshots 1,2,3,4,5, the reconstructed snapshot is very close to the given true snapshot. The boundary matching for the second case is not very accurate. In the next chapter, we will indicate how to place additional constraints for better boundary matching.

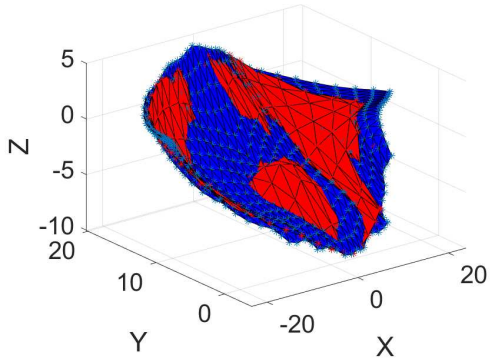
Table 5.2: Operator splitting method for diffeomorphisms from surface to surface

Case	Case 1	Case 2
Grid size for $\boldsymbol{x_0}$	144	288
Grid size for $\boldsymbol{y_0}$	576	576
Horizon length L	6	6
Fixed number of iterations	80	80
KKT solving time (ms)	1.91	3.73
Quadratic control step time (ms)	0.06	0.12
Single period proximal step time (ms)	0.84	2.33
Shape matching disparity	9×10^{-3}	6×10^{-3}
Number of Newton iteration	5	5

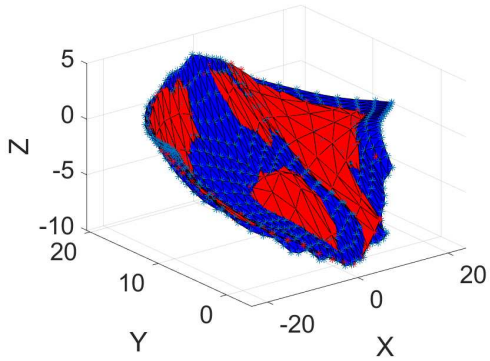


(a) Frame 0

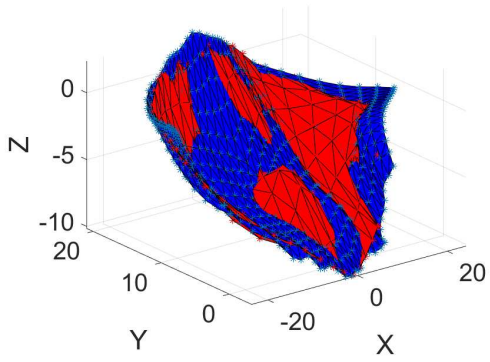
(b) Frame 1



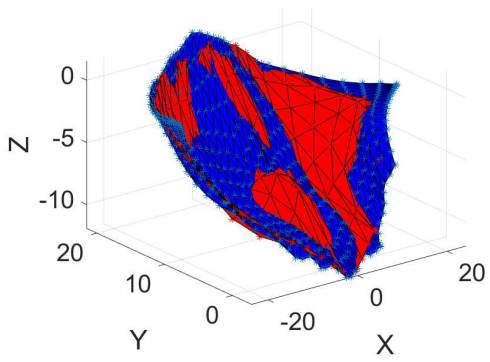
(c) Frame 2



(d) Frame 3

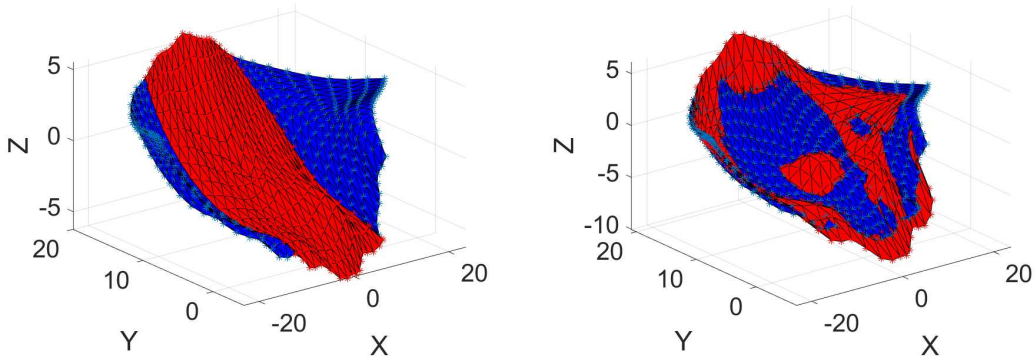


(e) Frame 4



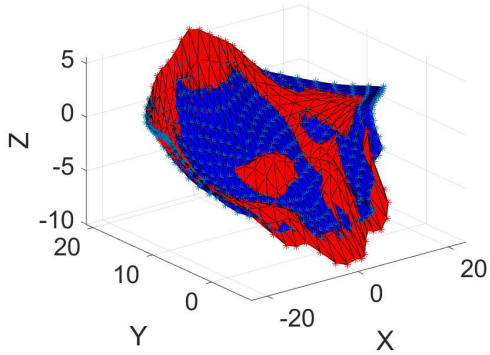
(f) Frame 5

Figure 5.2: Comparison of reconstructed snapshot (red) and target snapshot (blue) for six 3D image frames of case 1

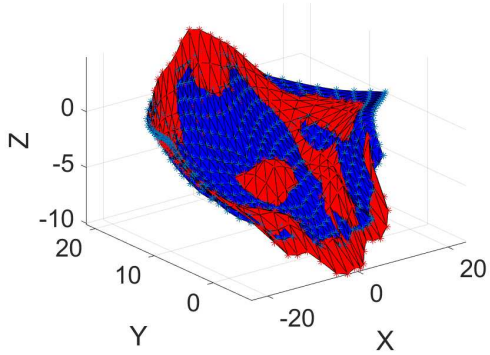


(a) Frame 0

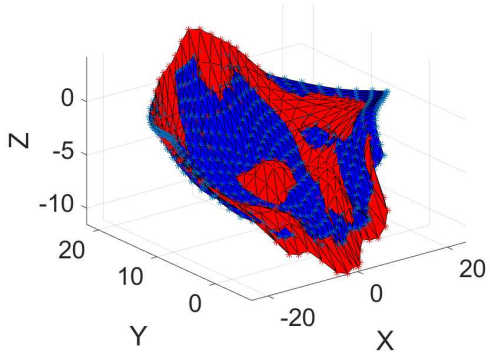
(b) Frame 1



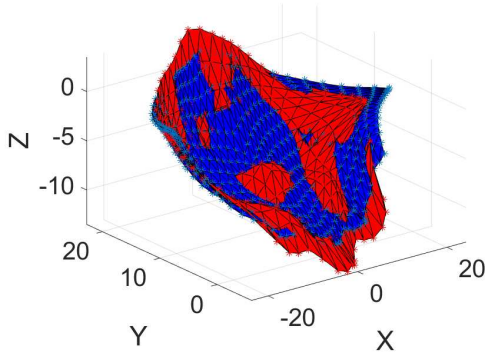
(c) Frame 2



(d) Frame 3



(e) Frame 4



(f) Frame 5

Figure 5.3: Comparison of reconstructed snapshot (red) and target snapshot (blue) for six 3D image frames of case 2

CHAPTER 6

Diffeomorphic Matching for Surfaces with Boundaries

6.1 Adjustment

In the cost functions studied precedingly, the weights in the matching disparity terms are identical for interior points and boundary points. In order to improve the boundary matching accuracy, a larger weight is added to the shape disparity terms involving boundary points. Specifically, denote by $\boldsymbol{x}'_{\boldsymbol{k}}$, $\boldsymbol{y}'_{\boldsymbol{k}}$ the boundary points for the reconstructed snapshot and true snapshot. Denote by N' and M' the respective grid sizes of the reconstructed and true boundaries. The cost function (4.1) and the

minimization problem are now modified as follows:

$$\begin{aligned} \text{minimize} \quad & J(\{\boldsymbol{\alpha_k}\}_{k=0}^{L}) := \sum_{k=0}^{L-1} \phi(\boldsymbol{x_k}, \boldsymbol{\alpha_k}; t_k) \\ & \qquad + \sum_{k=1}^{L} \lambda_k \psi(\boldsymbol{x_k}) + \sum_{k=1}^{L} \hat{\lambda}_k \hat{\psi}(\boldsymbol{x'_k}), \end{aligned} \tag{6.1}$$

$$\begin{aligned} \text{subject to} \quad & \boldsymbol{x_{k+1}} = \boldsymbol{x_k} + \tau_k \mathbb{K}(\boldsymbol{x_k})\boldsymbol{\alpha_k}, \quad k = 0, \cdots, L-1, \\ & \boldsymbol{x_0} = \boldsymbol{x}_{init}, \end{aligned} \tag{6.2}$$

where each $\hat{\lambda}_k$ is a new weight associated with the boundary matching term for the $k-th$ reconstructed and true snapshots. The three functionals ϕ, ψ, $\hat{\psi}$ are now defined as follows:

$$\phi(\boldsymbol{x_k}, \boldsymbol{\alpha_k}; t_k) = \frac{\tau_k}{2} \boldsymbol{\alpha_k}^T \mathbb{K}(\boldsymbol{y_k})\boldsymbol{\alpha_k}, \tag{6.3}$$

$$\begin{aligned} \psi(\boldsymbol{x_k}) = & \frac{1}{N^2} \sum_{i,j=1}^{N} K_s(x_i(t_k), x_j(t_k)) - \frac{2}{NM} \sum_{i=1}^{N} \sum_{j=1}^{M} K_s(x_i(t_k), y_j(t_k)) \\ & + \frac{1}{(M)^2} \sum_{i,j=1}^{M} K_s(y_i(t_k), y_j(t_k)), \end{aligned} \tag{6.4}$$

$$\begin{aligned} \hat{\psi}(\boldsymbol{x'_k}) = & \frac{1}{N'^2} \sum_{i,j=1}^{N'} K_{s'}(x'_i(t_k), x'_j(t_k)) - \frac{2}{N'M'} \sum_{i=1}^{N'} \sum_{j=1}^{M'} K_{s'}(x'_i(t_k), y'_j(t_k)) \\ & + \frac{1}{(M')^2} \sum_{i,j=1}^{M'} K_{s'}(y'_i(t_k), y'_j(t_k)). \end{aligned} \tag{6.5}$$

The operator splitting method to solve this slightly modified optimal control problem is given by:

$$(\boldsymbol{x}^{n+1}, \boldsymbol{\alpha}^{n+1}) := \underset{(\boldsymbol{x},\boldsymbol{\alpha})\in\mathcal{D}}{\arg\min}(\Phi(\boldsymbol{x}, \boldsymbol{\alpha}) + \frac{\rho}{2}||(\boldsymbol{x}, \boldsymbol{\alpha}) - (\tilde{\boldsymbol{x}}^n, \tilde{\boldsymbol{\alpha}}^n) - (z^n, \beta^n)||_2^2),$$

$$(\tilde{\boldsymbol{x}}^{n+1}, \tilde{\boldsymbol{\alpha}}^{n+1}) = \underset{(\tilde{\boldsymbol{x}},\tilde{\boldsymbol{\alpha}})}{\arg\min}(\Psi(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{\alpha}}) + \frac{\rho}{2}||(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{\alpha}}) - (\boldsymbol{x}^{n+1}, \boldsymbol{\alpha}^{n+1}) + (z^n, \beta^n)||_2^2),$$

$$(z^{n+1}, \beta^{n+1}) := (z^n, \beta^n) + (\tilde{\boldsymbol{x}}^{n+1}, \tilde{\boldsymbol{\alpha}}^{n+1}) - (\boldsymbol{x}^{n+1}, \boldsymbol{\alpha}^{n+1}).$$

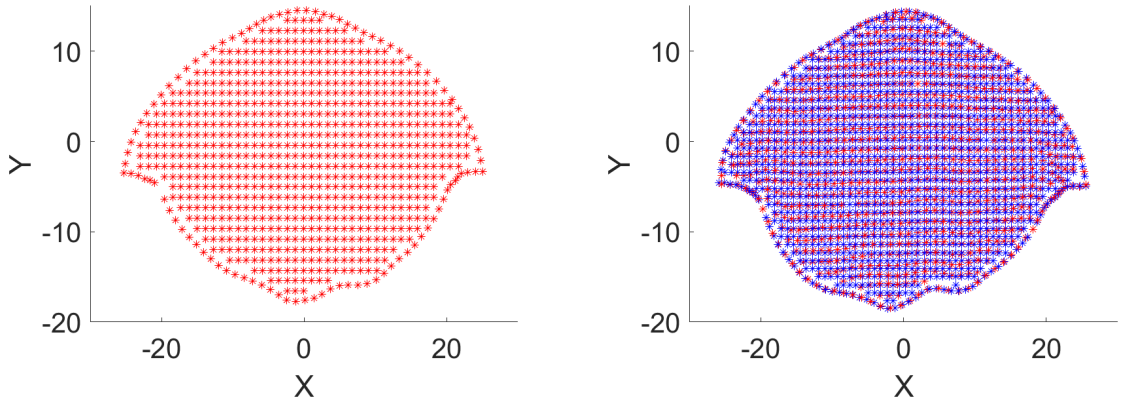
Here

$$\Psi(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{\alpha}}) = \sum_{k=1}^{L}(\lambda_k \psi(\tilde{\boldsymbol{x}}_{\boldsymbol{k}}) + \hat{\lambda}_k \hat{\psi}(\tilde{\boldsymbol{x}}'_{\boldsymbol{k}})). \tag{6.6}$$

There are no changes in the quadratic control step for the kinetic energy and the last step. The only thing that needs to be adjusted is the single period proximal step, because we use a different Ψ function.

6.2 Matlab Implementation

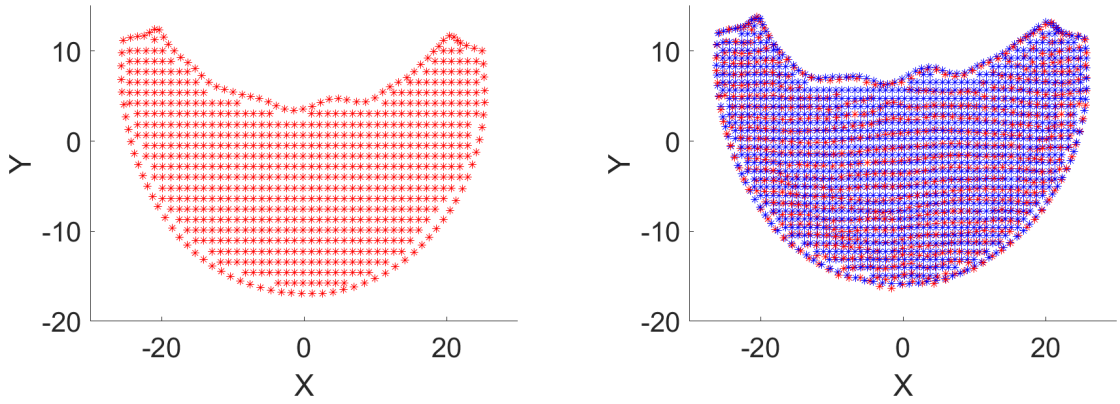
As discussed in Chapter 2, we first construct an almost uniform reconstruction of the discretizing point grids for the anterior leaflet and the posterior leaflet of a mitral valve. The results are shown in Figures 6.1 and 6.2, respectively. Figure (a) displays the discretizing point grid of the reconstructed initial surface, which is the midsystol snapshots. Figure (b) displays the comparison of discretized point grid of reconstructed target surface with TOMTEC discretized snapshots at endsystol.



(a) Anterior leaflet of new uniform discretizing point grid on midsystole frame

(b) Anterior leaflet of new uniform (red) with existing (blue) discretizing point grid on endsystole frame

Figure 6.1: Anterior leaflet



(a) Posterior leaflet of new uniform discretizing point grid on midsystole frame

(b) Posterior leaflet of new uniform (red) with existing (blue) discretizing point grid on endsystole frame

Figure 6.2: Posterior leaflet

6.2.1 Shape Matching

In Figures 6.3 and 6.4, we display the diffeomorphic deformation of the anterior and posterior leaflets. The bottom surface is the initial snapshot, and the top surface is the fifth snapshot.

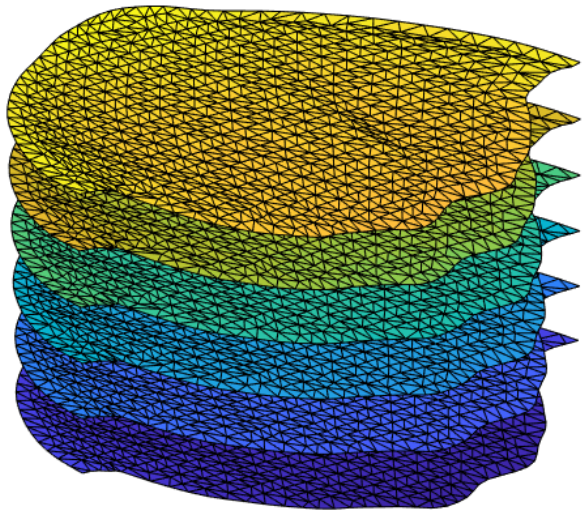


Figure 6.3: Successive deformations of anterior leaflet between midsystole and endsystole (six 3D image frames).

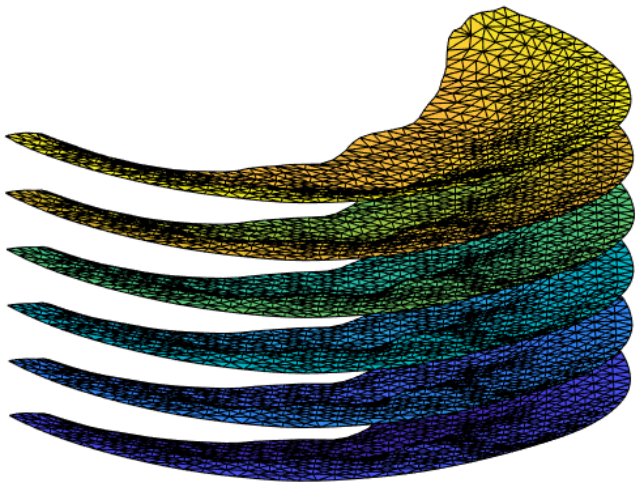
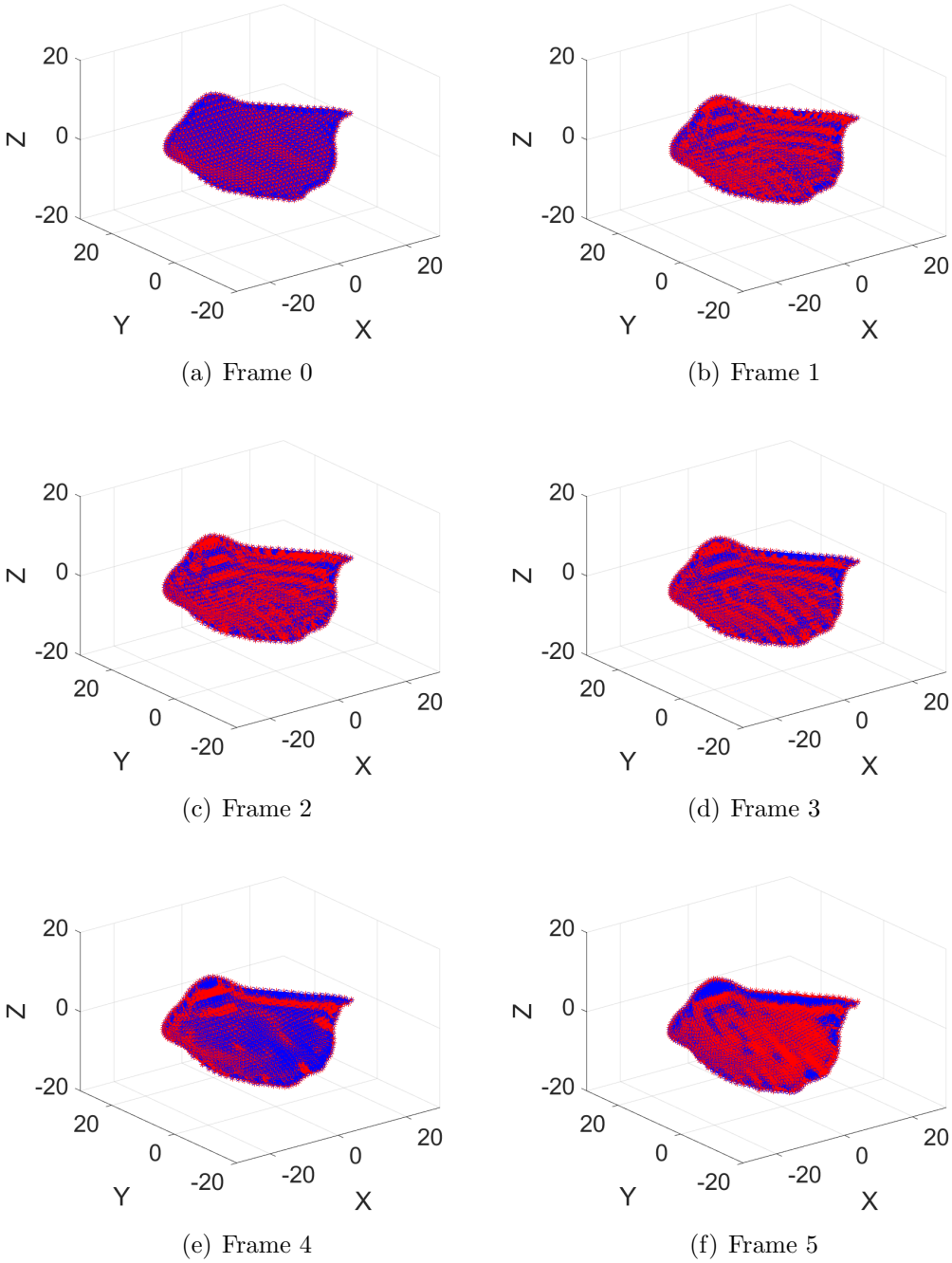


Figure 6.4: Successive deformations of posterior leaflet between midsystole and endsystole (six 3D image frames).

The comparison of our reconstructed snapshots with target snapshots are displayed in Figures 6.5 and 6.6, respectively. These new snapshots matching are much more accurate than the results obtained in the preceding chapter without introducing stronger constraint on boundary matching.



(a) Frame 0

(b) Frame 1

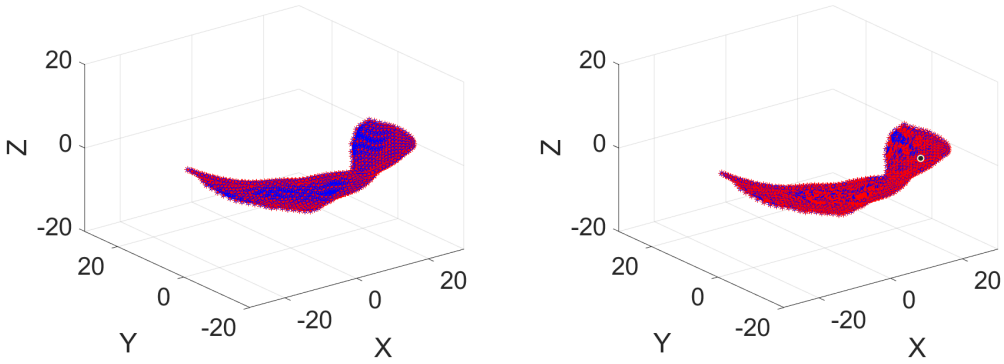
(c) Frame 2

(d) Frame 3

(e) Frame 4

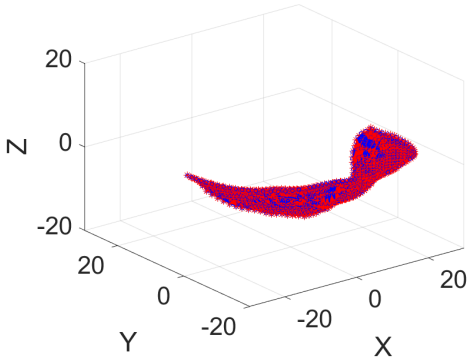
(f) Frame 5

Figure 6.5: Comparison of reconstructed snapshots (red) with target snapshots (blue) for the anterior leaflet.

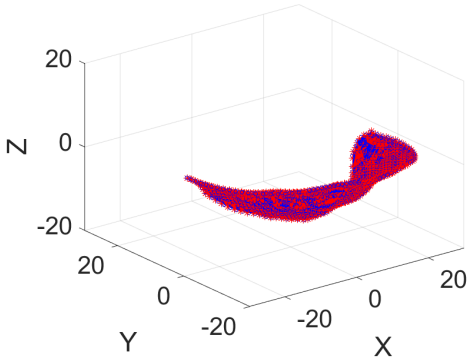


(a) Frame 0

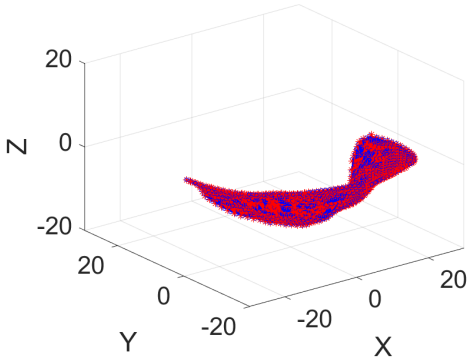
(b) Frame 1



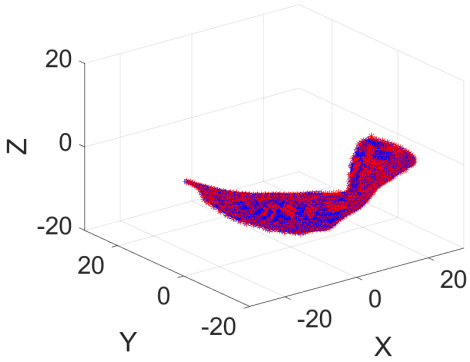
(c) Frame 2



(d) Frame 3



(e) Frame 4



(f) Frame 5

Figure 6.6: Comparison of reconstructed snapshots (red) with target snapshots (blue) for the posterior leaflet

At the end of the algorithmic cost minimization, the Hausdorff distances between the reconstructed and targeted snapshots are systematically computed to quantify the physical accuracy of the diffeomorphic reconstruction. The Hausdorff distance curves displayed in Figure 6.7 decrease to a stable value, which nearly matches the mesh size of our discretization grids.

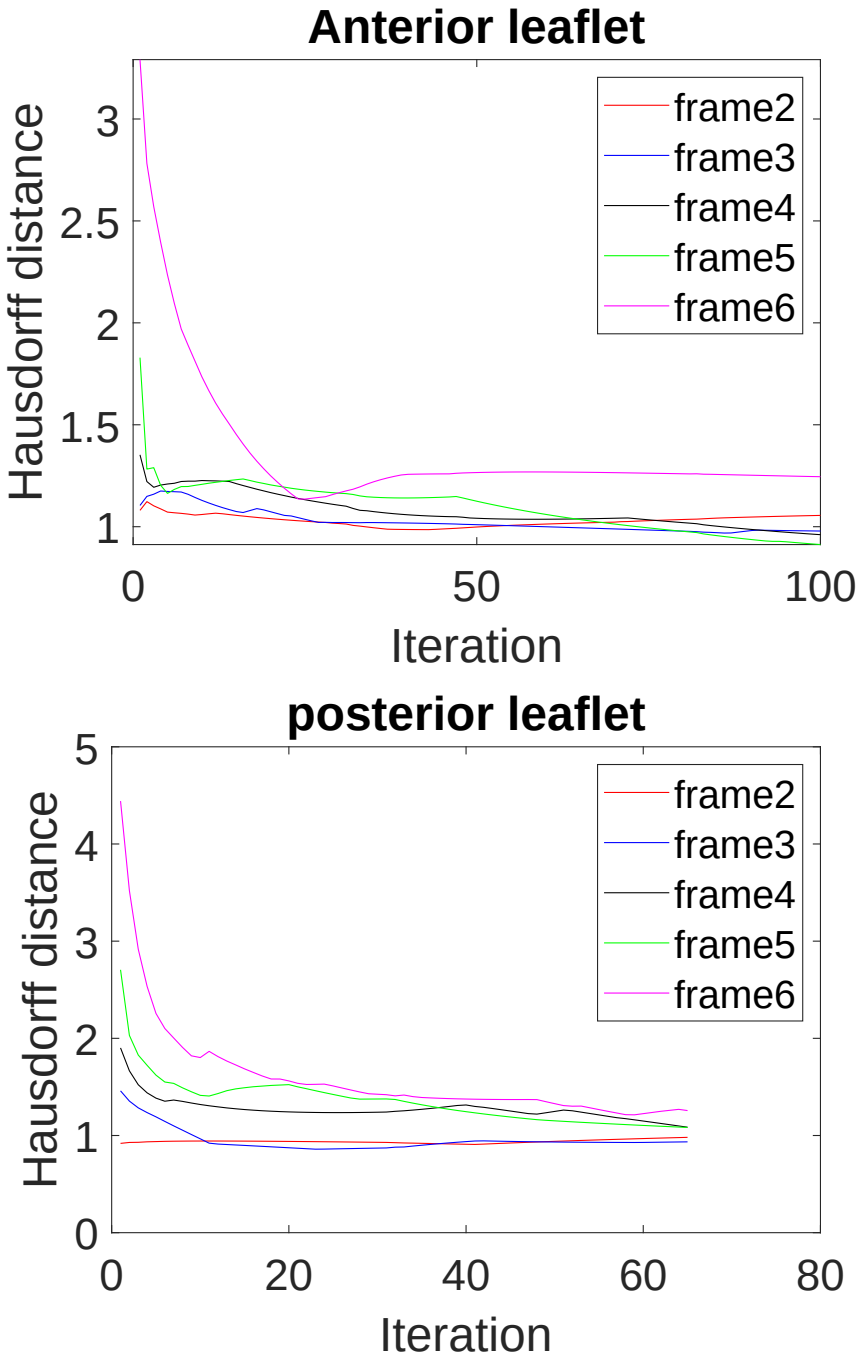


Figure 6.7: Hausdorff distances between reconstructed snapshots and target snapshots for each mitral valve leaflet

CHAPTER 7

Numerical Comparison between Operator Splitting and Second Order Newton Descent

In this chapter, we numerically compare our operator splitting method with the second order variational approach implemented in earlier studies by Yue Qin. In her dissertation [47], she used a second order Newton descent approach to construct an optimal diffeomorphic matching a given finite set of mitral valve leaflets snapshots. The basic idea in her work was to use Bellman's optimality principle with quadratic approximation to solve a nonlinear optimal control problem. We refer to her thesis [47] for details.

The continuous time nonlinear control problem is subjected to the following ODE constraints

$$\frac{dx}{dt} = F(x(t), \alpha(t)), \quad t \in (0, 1],$$
$$x(0) = x^0$$

where the cost associated to the trajectory $x(t)$ and the controls $\alpha(t)$ is given by

$$J(\alpha) = \int_0^1 g(x(t), \alpha(t))dt + \phi(x(1))$$

where the terminal payoff $\phi(x(1))$ is the disparity between $x(1)$ and the target set S^1, see (3.11).

In the classical Bellmann formalism, the associated value function $V(x, t)$ is then defined by

$$V(x, t) = \inf_{\alpha}(\int_t^1 g(x(t), \alpha(t))dt + \phi(x(1))) \tag{7.1}$$

The value function $V(x, t)$ represents the cost incurred from starting in state $x(t)$ at time t and controlling the system optimally from then until final time $t = 1$.

Based on Bellman's optimality principle, and assuming that the value function V is a C^1 function of the variable (x, t), one needs to solve the nonlinear partial differential equation

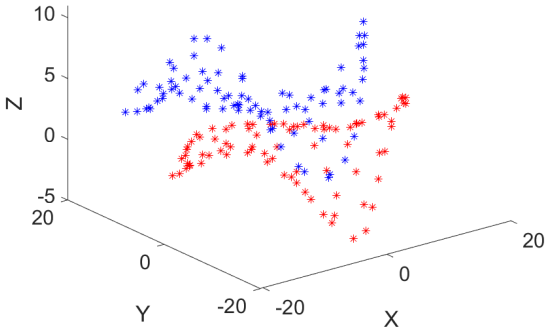
$$V_t(x, t) + \min_{\alpha}\{F(x, \alpha) \cdot \bigtriangledown_x V(x, t) + g(x, a)\} = 0$$

with final state condition for $x \in R^d$, $V(x,1) = \phi(x)$.

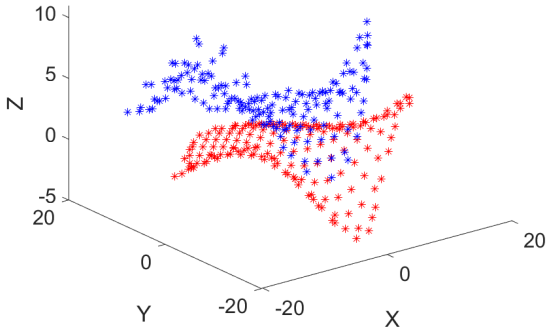
In [47], for the diffeomorphic matching problem, the authors introduce a quadratic approximation for all the functions involved, in order to numerically derive the feedback control law. To apply their approach to the diffeomorphic matching problem , a key technical point in [47] is the use of the second order Newton descent method.

We have tested and compared these two diffeomorphic matching techniques in four cases where we want to deform N discretized grid points of initial given surface (S^0) to a given M_1 discretized grid points of target surface (S^1) by a diffeomorphic matching. Figure 7.1 displays those grid points of surface. The results are displayed in Table 7.1. Our operator splitting method is more stable and much faster than the Newton descent method implemented by Yue Qin. When sizes of the point grids discretizing the surfaces become greater than 400 points, the Newton descent algorithm can often break down due to the singularity of the Hessian matrix.

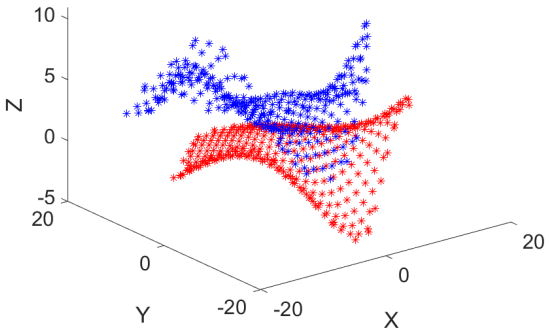
We select the fourth case to illustrate the comparison of matching disparity, kinetic energy, CPU time and Hausdorff distances in Figures 7.2, 7.3, 7.4 and 7.5. 7.4 displays the statistical quantitative measure of distance $\mathcal{D}(x, S^1)$ in (3.9) with $x \in \hat{S}^1$ for every iteration.



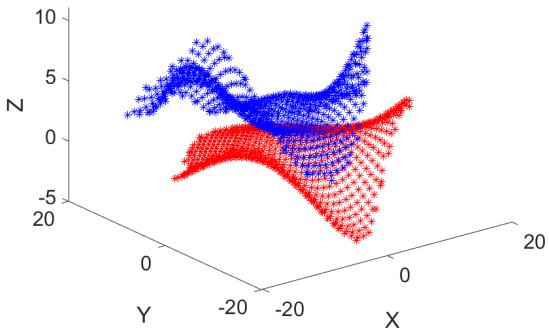
(a) Case 1: 83 grid points of initial surface (red) and 82 grid points of target surface (blue).



(b) Case 2: 172 grid points of initial surface (red) and 160 grid points of target surface (blue).



(c) Case 3: 332 grid points of initial surface (red) and 302 grid points of target surface (blue).

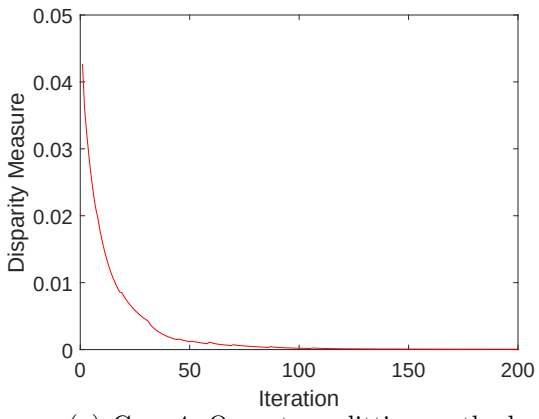


(d) Case 4: 657 grid points of initial surface (red) and 654 grid points of target surface (blue).

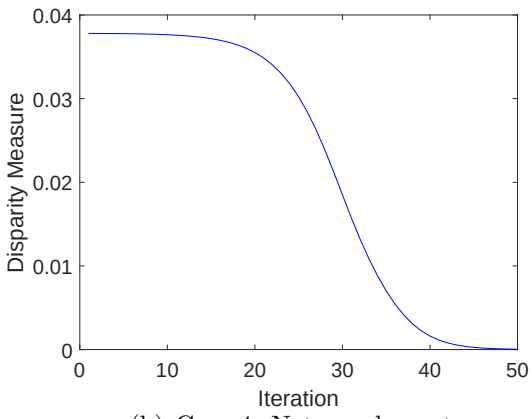
Figure 7.1: Four cases of discretized grid points of initial given surface (red) and target surface (blue).

Table 7.1: Comparison of operator splitting method with Newton descent method. N and M_1 are the grid sizes for the initial snapshot and target snapshot. $L = 2$ is the number of given snapshots

Case	Parameter	Operator splitting	Newton descent
Case 1	Number of iterations	200	52
$N : 83$	Matching disparity	2.6×10^{-3}	4.2×10^{-5}
$M_1 : 82$	Hausdorff distance	3.13	2.53
$L : 2$	Kinetic energy	186	243
	CPU time/iteration	0.023	0.073
Case 2	Number of iterations	200	49
$N : 172$	Mmatching disparity	4.91×10^{-4}	4.06×10^{-5}
$M_1 : 160$	Hausdorff distance	1.80	2.39
$L : 2$	Kinetic energy	345	152
	CPU time/iteration	0.086	0.401
Case 3	Number of iterations	200	51
$N : 332$	Matching disparity	3.37×10^{-4}	4.47×10^{-5}
$M_1 : 302$	Hausdorff distance	1.60	2.88
$L : 2$	Kinetic energy	616	343
	CPU time/iteration	0.36	1.81
Case 4	Number of iterations	200	50
$N : 657$	Matching disparity	5.46×10^{-5}	4.10×10^{-5}
$M_1 : 654$	Hausdorff distance	1.11	3.22
$L : 2$	Kinetic energy	1179	327
	CPU time/iteration	1.87	9.9

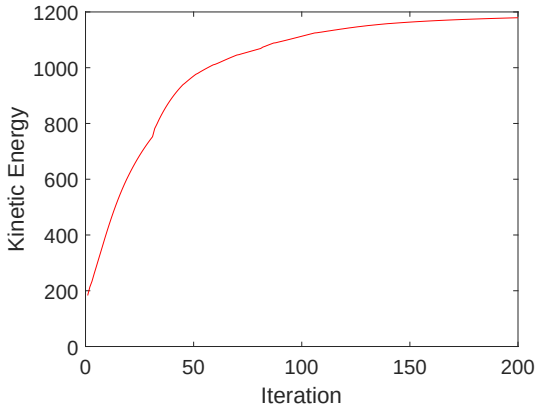


(a) Case 4: Operator splitting method

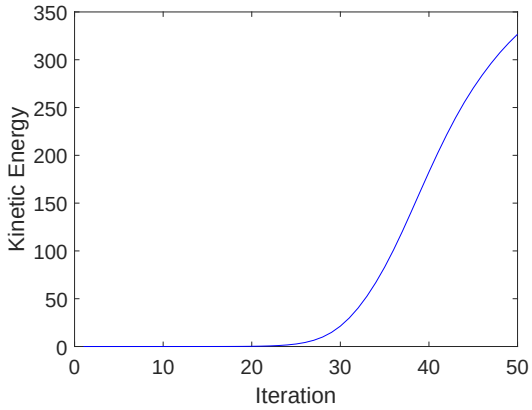


(b) Case 4: Netwon descent

Figure 7.2: Matching disparity

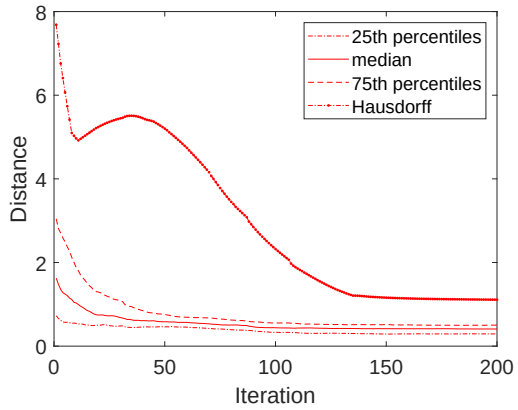


(a) Case 4: Operator splitting method

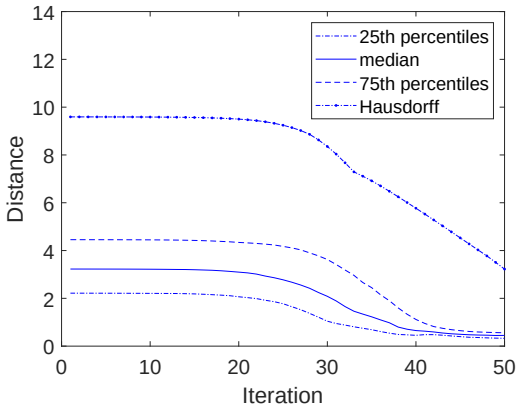


(b) Case 4: Newton descent

Figure 7.3: Kinetic energy

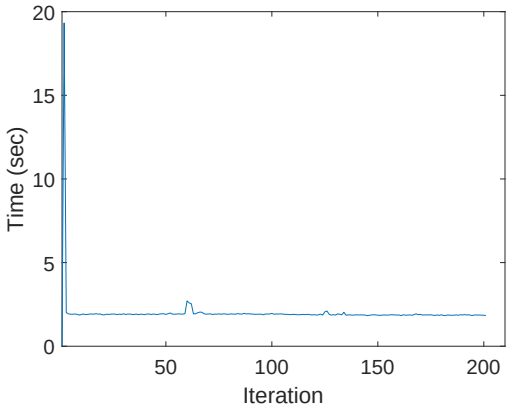


(a) Case 4: Operator splitting method

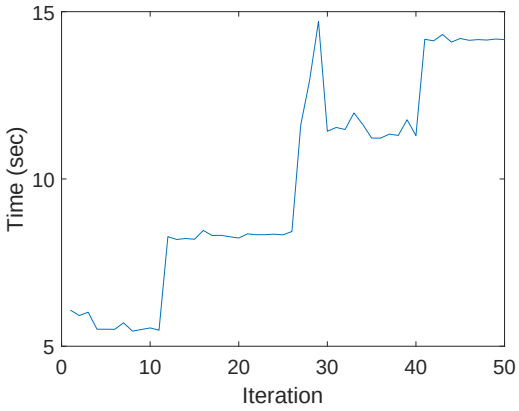


(b) Case 4: Newton descent

Figure 7.4: Distances comparison



(a) Case 4: Operator splitting method



(b) Case 4: Newton descent

Figure 7.5: CPU time per iteration in seconds

CHAPTER 8

Strain Intensities: Definition and Computation

8.1 Strain Intensities: Definition

Around each point p of the mitral valve observed at midsystole, we characterize "local tissue fatigue" by "local tissue length deformation" around p between midsystole and endsystole. To quantify this local length deformation, we compute and analyze the strain tensor at p for all mitral valve grid points. This is a technically sophisticated goal, for which we have developed a highly efficient numerical approach. The main output of strain tensor computation and analysis is the numerical derivation of an intuitively interpretable strain intensity SI(p) at p. Strain intensity SI(p) can be

practically viewed as quantifying relative tissue length deformation around p.

This intuitive interpretation of $\mathrm{SI}(p)$ is roughly valid for most small mitral valve tissue segments around the point p. Since relative deformation of length can be viewed as a percentage, each value $\mathrm{SI}(p)$ will be a percentage, a value varying between 0% and 100%.

8.2 Concrete Computation of Strain Intensities

8.2.1 Length Ratio Method

We need to compute the average geometric strain around each point p, which is called the geometric strain intensity $\mathrm{S}(p)$ which we now define.

Call GMS and GES the point grids discretizing a given mitral valve leaflet at midsystole and endsystole respectively. Assume that we have already computed a diffeomorphic matching $F : R^3 \to R^3$ such that $F(GMS) = GES$. For each $p \in GMS$, define a small neighborhood $\mathcal{N}(p)$ of p, with discrete radius $r(p)$, as the set of the k points of GMS which are closest to p. For instance, we can take $k = 6$ or $k = 12$ or some other value adapted to the mesh size of GMS; the important constraint is that these points should be physically close to $p \in R^3$, and this should be checked numerically. Then for every point $q \in \mathcal{N}(p)$, define the corresponding length deformation by the deformation ratio

$$\mathcal{R}(p, q) = D(p', q')/D(p, q) \tag{8.1}$$

where $p' = F(p)$ and $q' = F(q)$ are the points of GES associated with points p and q by the deformation F. Here $D(p, q)$ is the distance between p and q on the discretized surface GMS, and $D(p', q')$ is similarly defined on GES. For faster implementation when the mesh sizes of GMS and GES are fairly small, we can reasonably approximate $D(p, q)$ and $D(p', q')$ by Euclidean distances in R^3. The strain $S(p)$ at p will then be computed by the dimensionless number

$$S(p) = 1/k \sum_{q \in \mathcal{N}(p)} \mathcal{R}(p, q) \tag{8.2}$$

which is the local average multiplicative factor representing average length deformation around the point p, between midsystole and endsystole.

When $S(p) > 1$, we have local surface dilation around p between midsystole and endsystole. When $S(p) < 1$, we have local surface contraction around p. Thus large values of $|S(p) - 1|$ are indicative of potential high leaflet tissue fatigue at p. So the local strain intensity $\mathrm{SI}(p)$ is defined as follows:

$$\mathrm{SI}(p) = |S(p) - 1| \tag{8.3}$$

8.2.2 Eigenvalue Method

The mean length ratio characterization of local strain which we have just described is easily implemented and very efficient, but we have also tested and implemented a second approach, much closer to the classical theoretical definition of the strain tensor. Consider only two snapshots of a specific mitral leaflet, namely the midsystole

snapshot L^0 at time $t = 0$ and the endsystole snapshot L^1 at time $t = 1$. We fix $k = 8$ to be the number of closest neighbors of p and $p' = F(p)$ to define $\mathcal{N}(p)$ and $\mathcal{N}(p')$. We will denote $p = p^0$ and $p' = p^1$ in the following definitions.

Call $T(p^0)$ the local tangent plane to L^0 at the point p^0. We first have to correctly approximate the unit normal vector to $T(p^0)$. For each triangle $\triangle pab$, where $a, b \in \mathcal{N}(p)$, the normal vector to $\triangle pab$ is the cross product $\overrightarrow{pa} \times \overrightarrow{pb}$.

A properly weighted average of all these vectors computed over all pairs a, $b \in \mathcal{N}(p)$ provides us with a good approximation of the unit normal vector to $T(p^0)$, and hence with a good approximation of $T(p^0)$, which we still denote by $T(p^0)$ for brevity.

The neighborhood $\mathcal{N}(p^0)$ is then projected onto the just constructed planar approximation of $T(p^0)$.

Similarly we compute $T(p^1)$ and we project $\mathcal{N}(p^1)$ onto p^1. As for p^0 we then denote $\mathbf{\Pi}$ the projection onto $T(p^1)$, and we define $\mathbf{\Pi}\mathcal{N}^1 = \mathbf{\Pi}(\mathcal{N}(P^1))$. To compute the local 2×2 surface strain tensor associated with the diffeomorphic deformation F at p^0, we use planar coordinates $q = (x, y)$ in $T(p^0)$ for each point $q = \mathbf{\Pi}(p)$ with p in $\mathcal{N}(p^0)$ and planar coordinates $q' = [W_1(x, y), W_2(x, y)] \in T(p^1)$ for the corresponding point $q' = \mathbf{\Pi}(F(q))$. We will using second order polynomials to approximate functions.

We can then approximate explicitly the classical local surface strain tensor by

the matrix

$$\mathcal{J}(p) = \begin{pmatrix} \frac{\partial W_1(x,y)}{\partial x} & \frac{1}{2}(\frac{\partial W_1(x,y)}{\partial y} + \frac{\partial W_2(x,y)}{\partial x}) \\ \frac{1}{2}(\frac{\partial W_2(x,y)}{\partial x} + \frac{\partial W_1(x,y)}{\partial y}) & \frac{\partial W_2(x,y)}{\partial y} \end{pmatrix}.$$

The geometric tissue strain $geoStr(p)$ at p is the square root of the ratio of area between deformed and reference small tissue patches around p, which is an approximation of the average deformation ratio of small tissue lengths around p between midsystole and endsystole. The specific formula is

$$geoStr(p) = \sqrt{|\det(\mathcal{J}(p))|} = \sqrt{|\lambda_1||\lambda_2|}, \tag{8.4}$$

where λ_1, λ_2 are the eigenvalues of the local surface strain tensor matrix $\mathcal{J}(p)$ with corresponding eigenvectors v_1 and v_2. When $|\lambda| > 1$, we have a local surface dilation around p in direction v. When $|\lambda| < 1$, similarly, we have a local surface contraction. The leaflet strain intensity $\mathrm{SI}(p)$ at p is then given by

$$\mathrm{SI}(p) = |geoStr(p) - 1|. \tag{8.5}$$

We can also consider the biggest difference of the eigenvalue with 1, which is

$$\mathrm{SI}(p) = \max(||\lambda_1| - 1|, ||\lambda_2| - 1|). \tag{8.6}$$

8.3 Quantile Curves of Strain Intensites

For each leaflet, at midsystole, our numerical codes compute the empirical quantiles of strain intensities values and display them as a quantile curve. Recall that si is the qv quantile for the variable SI, if $\Pr[\text{SI} < si] = qv$ or equivalently $\Pr[\text{SI} \geq si] = 1-qv$. For each leaflet the quantile $Q(50\%)$ is its median strain. Extreme quantiles such as $Q(0\%)$ or $Q(100\%)$ are deliberately not displayed because they are by nature definitely less accurate.

Note that a leaflet with overall high strain intensities (such as a prolapsed posterior leaflet), is typically characterized by higher quantile curves, as will be observed in the prolapse case.

8.4 Graphic Displays of Strain Intensites

Strain intensities $\text{SI}(p)$ are computed here for all the interior grid points p of the anterior leaflet, as well as for the posterior leaflet with max eigenvalue. These two spatial distributions of strain intensities over the two leaflets observed at mid-systole, are then automatically displayed in 3D by numerical codes via a color map, assigning to each leaflet grid point p a color which codes its strain intensity $\text{SI}(p)$, with deep blue indicating low strain and deep red indicating high strain. Figure 8.1 displays graphically the strain intensities values for a given mitral valve. Figure 8.2 displays separately the two strain quantile curves for the anterior and posterior leaflets. The left side of Figure 8.2 displays strain quantile curves computed by the length ratio

method, and the right side of Figure 8.2 is computed by evaluating the the local surface strain tensors. The strain values estimated by the length ratio method tend to be slightly larger than those computed via estimation of the surface strain tensors.

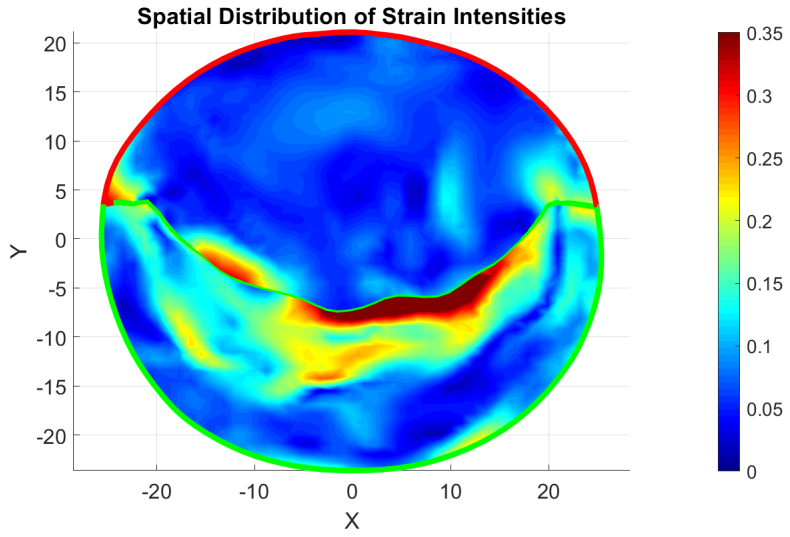
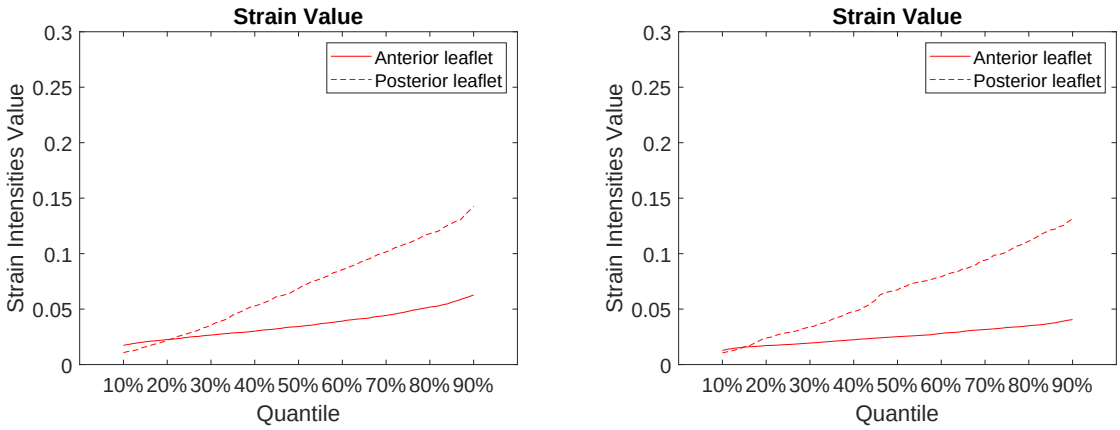


Figure 8.1: Spatial distribution of strain intensities on the midsystole mitral valve leaflets

Figure 8.3 displays the contraction strain intensity $||\lambda| - 1|$ region with $|\lambda| < 1$ and the dilation strain intensity $||\lambda| - 1|$ region with $|\lambda| > 1$. Figure 8.4 displays the eigenvectors of local strain tensor.

A comparative statistical analysis of strain distributions due to mitral valve dynamic deformations was first published in [33], to compare normal patients with patients having primary or secondary mitral regurgitation. One conclusion in [33] is that primary mitral regurgitation patients tend to have higher mitral valve strain. Strain quantile curves for secondary mitral regurgitation are roughly comparable to

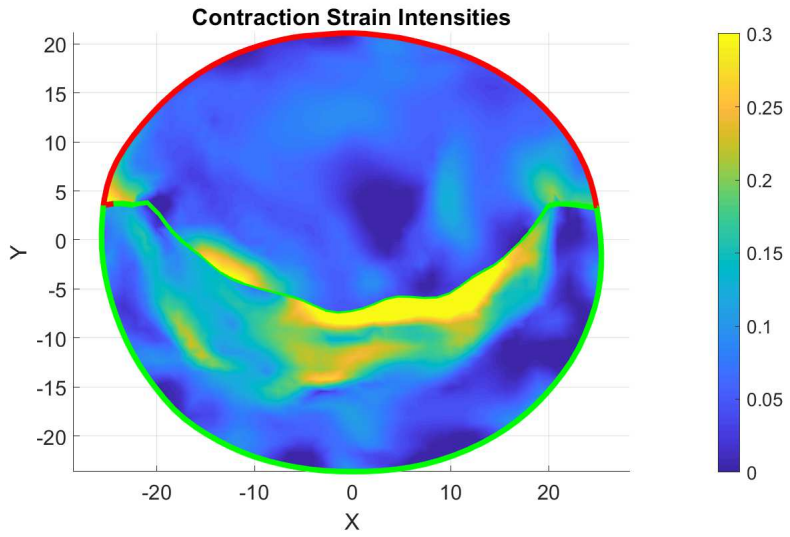


(a) Strain value by length ratio method

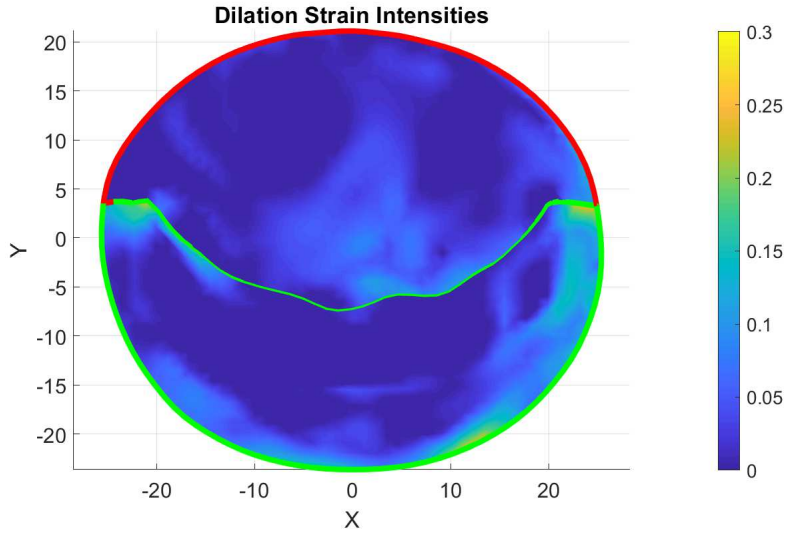
(b) Strain value by eigenvalue method

Figure 8.2: Strain value quantile curve

strain quantile curves for normal patients. Mitral valve strain tends to be higher for the posterior leaflet compared to the anterior leaflet, and this result hold for normal patients as well as for patients with mitral regurgitation. Numerical strain analysis from echocardiographic 3D movies has the potential to improve characterization of mitral valve diagnosis and to quantify the strain reduction impact of mitral valve surgery.

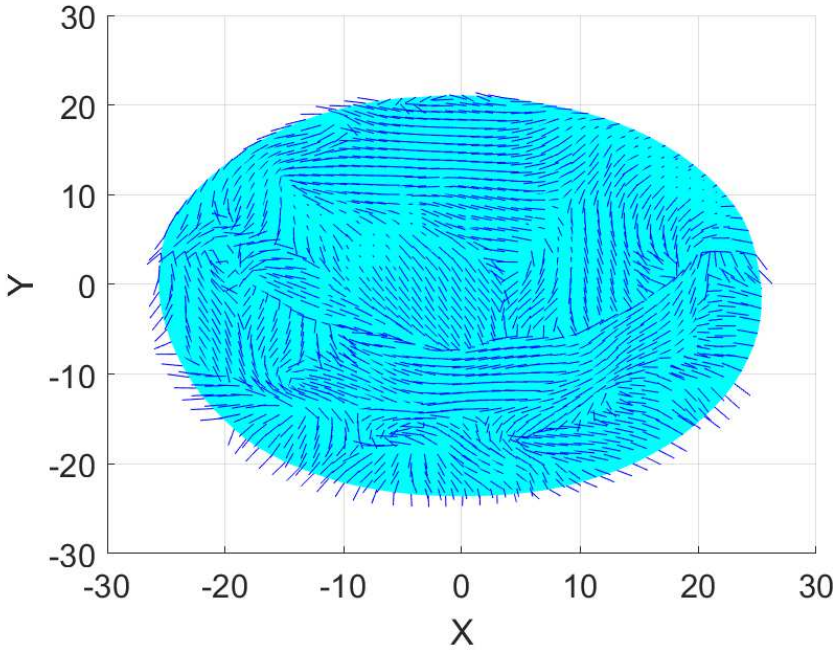


(a) Contraction strain intensity

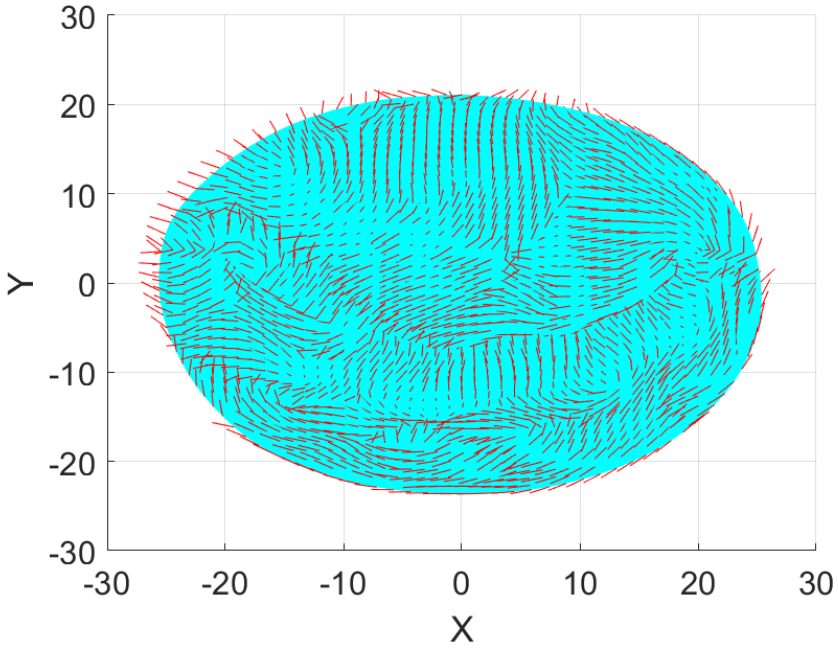


(b) Dilation strain intensity

Figure 8.3: Spatial distribution of strain intensities with contraction and dilation region.



(a) Spatial distribution of eigenvector v_1



(b) Spatial distribution of eigenvector v_2

Figure 8.4: Spatial distribution of eigenvector.

CHAPTER 9

Mitral Valve Segmentation

The mitral valve consists of two deformable leaflets, the anterior leaflet (Ant) and the posterior leaflet (Pos), which together have a surface $4 - 6cm^2$. The two leaflets in this study are partitioned into a total of six scallops: $A1, A2, A3$ (anterior leaflet) and $P1, P2, P3$ (posterior leaflet). Relying on previous results of Jeff Freeman (PhD thesis [18]), we implemented these two segmentations based on the three scallops areas. The areas of $A1, A2, A3$ will respectively be 25%, 50% and 25% of the area of the anterior leaflet, with a similar constraint for $P1, P2, P3$ and the posterior leaflet. Specifically to construct the scallops $A1, P1$, we first choose two points along the annulus, one point Ana on the annulus boundary of the anterior leaflet, the other

point Anp on the annulus boundary of the posterior leaflet; we then connect the points Ana and Anp by a curve lying on the surface. If the areas of both A1 and P1 are close to 25% of the anterior leaflet area, we will fix the points Ana and Anp; otherwise, we will start an easy iterative adjustment for the positions of these two points in order to reach the desired values for the areas of A1 and P1. We also construct an annulus band Z_{ann} of radius r_{ann} around the annulus boundary of the anterior leaflet, with an easy iterative adjustment of r_{ann} to ensure that Z_{ann} contains 20% of the grid points of the anterior leaflet. We proceed similarly to generate a coaptation band Z_{coapt} of radius r_{coapt} around the coaptation boundary of the anterior leaflet, again with relative area close to 20%. The boundary zone Z_{bound} is defined as the union of the annulus band and the coaptation band. The complement of the boundary zone will be called the center zone Z_{center}. Finally, let $com1$ and $com2$ be the two commissure curves, which connect the annulus and the two angular points of the coaptation line. Then denote $COM = com1 \cup com2$. Select a radius r such that the region Z_{comm} of all leaflet points within distance r of COM covers an area close to 8% of the total leaflet area. Z_{comm} will be called the commissure zone. Figure 9.1, 9.2, 9.3 display all these regions on the mitral valve surface at midsystole.

For each patient, we have thus systematically identified 18 anatomically significant sub-regions of the mitral valve surface at midsystole, namely the 18 regions denoted Ant, Pos, A1, A2, A3, A_{ann}, A_{coap}, A_{center}, A_{bound}, A_{comm}, P1, P2, P3, P_{ann}, P_{coap}, P_{center}, P_{bound}, P_{comm}.

In Chapter 8, we have indicated how one can compute the strain intensity values for all grid points of the discretized mitral valve surface observed at midsystole.

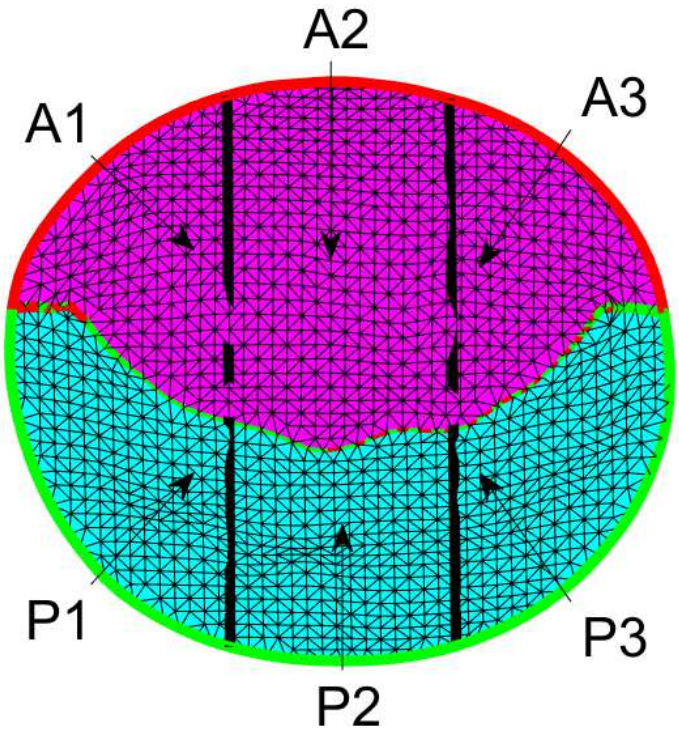


Figure 9.1: Scallops: A1, A2, A3, P1, P2, P3

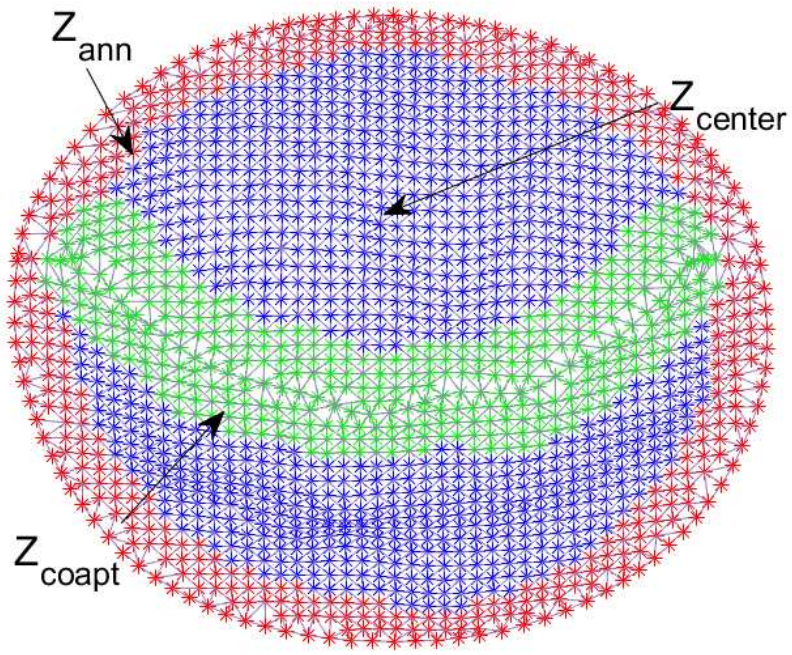


Figure 9.2: Annulus, coaptation, center zones

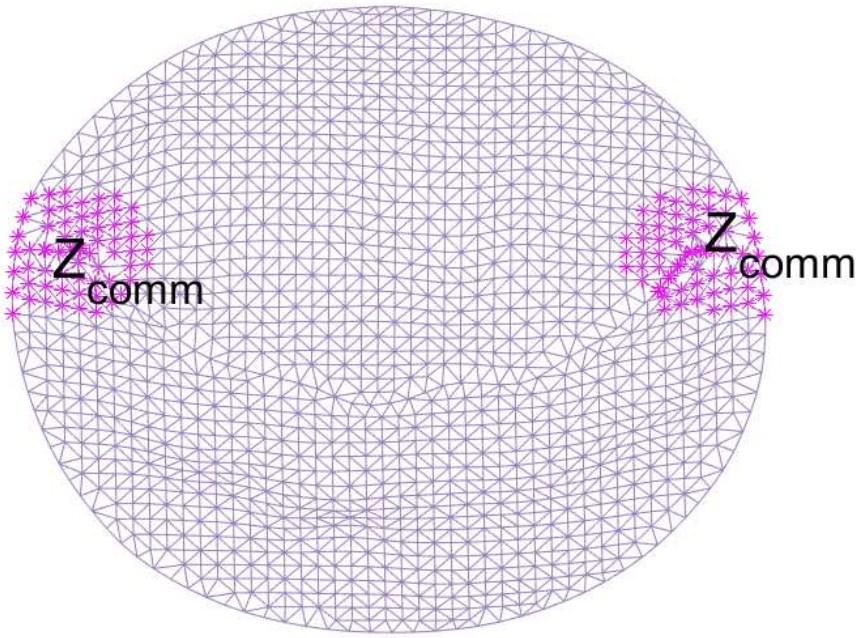


Figure 9.3: Commissure zone

For each one of the preceding 18 subregions, we calculate 9 quantile values, namely $Q(10\%)$, $Q(20\%)$, $\cdots$, $Q(90\%)$. We have systematically applied a software to test a large set of 159 mitral valves acquired at The Methodist Hospital (Cardiology). This intensive computational approach has generated $162 = 18 \times 9$ strain features for each one of the 159 mitral valve patients.

CHAPTER 10

Automatic Classification of Mitral Valve Patients by Machine Learning

In this chapter, we implement and test a support vector machines (SVM) approach to test the impact of strain distribution analysis for automatic discrimination between two small groups of patients, namely normal patients versus patients diagnosed with secondary regurgitation. For each patient i, denote by x_i the 162 dimensional feature vector. Denote by $y_i \in \{-1, +1\}$ the binary classification of patient i as a normal case or a regurgitation case.

10.1 Support Vector Machines (SVM)

Denote by TR our training data set, consisting of N pairs (x_1, y_1), (x_2, y_2),$\cdots$, (x_N, y_N), with $x_i \in R^p$ and $y_i \in \{-1, 1\}$. Denote a generic hyperplane $H(f)$ in R^p as the set of all x in R^p such that

$$f(x) = x^T\beta + \beta_0 = 0.$$

Recall that for any affine function f, and any z in R^p, $f(z)$ is the algebraic distance from point z to the hyperplane $H(f)$. The SVM binary classification seeks an optimal hyperplane separating the data points into two classes. For a linearly separable pair of classes, the optimal hyperplane $H(f)$ maximizes a margin defined as the distance between the two smallest half spaces having boundaries parallel to $H(f)$ and resp. containing the true positive and negative classes. Ideally, in the linearly separable situations, one seeks to find an affine function $f(x) = x^T\beta+\beta_0$ such that $y_i f(x_i) > 0$, $\forall i$. The SVM approach is then to solve the optimization problem

$$\max_{\beta,\beta_0,||\beta||=1} M,$$
$$\text{subject to } \; y_i(x_i^T\beta + \beta_0) \geq M, i = 1, \cdots, N.$$

The basic idea originally introduced by Vapnik [51] is to create the biggest margin $(2M)$ between the training points respectively belonging to class 1 and -1. A more

convenient way to formulate this problem is the following

$$\min_{\beta,\beta_0} ||\beta||,$$
$$\text{subject to} \quad y_i(x_i^T\beta + \beta_0) \geq 1, i = 1, \cdots, N,$$

here $M = 1/||\beta||$. This is a convex optimization problem. One classically uses the method of Lagrange multipliers to get an expression for β. Substituting β into a Lagrangian functional, we obtain the so-called Wolfe dual [52], a simpler convex optimization problem, where the Λ_i are Lagrange multipliers to be determined,

$$\mathcal{L}(\beta, \Lambda) = \frac{1}{2}||\beta||^2 - \sum_{i=1}^{N} \Lambda_i[y_i(x_i^T\beta + \beta_0) - 1].$$

Setting derivatives to zero, we obtain:

$$\beta = \sum_{i=1}^{N} \Lambda_i y_i x_i,$$

and the Wolfe dual problem becomes the minimization of

$$\mathcal{L}_D(\Lambda) = \sum_{i=1}^{N} \Lambda_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{k=1}^{N} \Lambda_i\Lambda_k y_i y_k x_i^T x_k, \tag{10.1}$$
$$\text{subject to} \quad \Lambda_i \geq 0.$$

The solution vector β is a linear combination of the support vectors which are the data points which lie on the hyperplane boundaries of the optimized margin zone.

For nonlinearly separable classes, the objective is the same, but to deal with the data points which are on the wrong side of their class boundary, one has to impose complementary adequate penalties. Introduce a slack variable ξ_i to penalize the wrongly classified data points. So one sets $\xi_i = 0$ for well classified data points, and $\xi_i \geq 0$ otherwise. The new constraint is $y_i(x_i^T\beta + \beta_0) \geq M(1-\xi_i), \forall i, \xi_i \geq 0, \sum_{i=1}^N \xi_i \leq$ constant. The basic idea of the formulation is to minimize the sum of penalties $\sum \xi_i$. Intuitively, one normalizes penalties so that misclassification occurs when $\xi_i > 1$. The minimization problem then becomes

$$\begin{aligned} &\min_{\beta,\beta_0,\xi} ||\beta||, \\ \text{subject to } \; &y_i(x_i^T\beta + \beta_0) \geq M(1-\xi_i), \quad \xi_i \geq 0, \quad \sum_{i=1}^N \xi_i \leq \text{constant}. \end{aligned}$$

Computationally, it is convenient to re-express this problem as follows

$$\begin{aligned} &\min_{\beta,\beta_0} \{||\beta||^2 + C\sum_{i=1}^N \xi_i\}, \\ \text{subject to } \quad &y_i(x_i^T\beta + \beta_0) \geq M(1-\xi_i), \xi_i \geq 0, \forall i, \end{aligned}$$

where the positive parameter C has to be optimally selected at a later stage. The Lagrangian becomes

$$\mathcal{L}(\beta, \Lambda, \Gamma) = \frac{1}{2}||\beta||^2 + C\sum_{i=1}^N \xi_i - \sum_{i=1}^N \Lambda_i[y_i(x_i^T\beta + \beta_0) - M(1-\xi_i)] - \sum_{i=1}^N \Gamma_i\xi_i. \quad (10.2)$$

Setting derivatives to zero, we get

$$\beta = \sum_{i=1}^{N} \Lambda_i y_i x_i.$$

After substituting this expression back into the Lagrange function, we obtain the dual objective function to maximize under constraints:

$$\mathcal{L}_D(\Lambda) = \sum_{i=1}^{N} \Lambda_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{k=1}^{N} \Lambda_i \Lambda_k y_i y_k x_i^T x_k, \tag{10.3}$$

$$\text{subject to } \ 0 \leq \Lambda_i \leq C, \sum_{i=1}^{N} \Lambda_i y_i = 0.$$

Maximizing the dual (10.3) is a simpler convex quadratic programming problem, and can be solved by standard techniques.

The support vector classifier described so far generates linear boundaries in the input feature space. Introduce a Hilbert space U of functions $g(x)$ defined on the feature space R^p. One can map the feature space into U by a nonlinear mapping h, so that optimal linear separation in U will then generate optimal nonlinear separation in the original feature space. In the Hilbert space U, select M functions $h_m(x), m = 1, \cdots, M$, and then define new input features by $h(x_i) = (h_1(x_i), h_2(x_i), \cdots, h_M(x_i))$, $i = 1, \cdots, N$. The nonlinear separator between our two classes will be of the form $f(x) = h(x)^T \beta + \beta_0$.

Proceding as above to get an optimal margin linear separator in the Hilbert space

U, one has then to minimze the objective functional

$$\mathcal{L}_D(\Lambda) = \sum_{i=1}^{N} \Lambda_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{k=1}^{N} \Lambda_i \Lambda_k y_i y_k < h(x_i), h(x_k) >, \tag{10.4}$$

where the inner product $<,>$ denotes the scalar product in U, and one must then have

$$\beta = \sum_{i=1}^{N} \Lambda_i y_i h(x_i).$$

So the solution function $f(x)$ can be written

$$\begin{aligned} f(x) &= h(x)^T \beta + \beta_0 \\ &= \sum_{i=1}^{N} \Lambda_i y_i < h(x), h(x_i) > + \beta_0. \end{aligned}$$

We now fix a positive definite kernel $K(x, x')$ defined for $x, x' \in R^p$, and define U to be the self-reproducing kernel Hilbert space defined by K. The space U is then generated by the functions $K_y(x) = K(y, x)$ with y arbitrary in R^p, and one has the classical self reproducing property $< K_y, K_z >_U = K(y, z)$, which is covered in Chapter 3.

Recall the well known theorem (see [53])

Theorem 10.1.1. *A positive definite kernel generates a reproducing Hilbert space, conversely a reproducing Hilbert space defines a positive kernel.*

Two popular choices for the kernel K in the SVM applications are the following:

$$\text{Polynomial kernel: } K(x, x') = (1+ < x, x' >)^d,$$

$$\text{Gaussian kernel: } K(x, x') = \exp(-r||x - x'||^2).$$

We had selected above arbitrary functions h_1, h_2, ... $h_N \in U$. One can now specify this choice more efficiently after selecting for U the self-reproducing Hilbert space associated with the positive definite kernel K. The dual function to be maximized then becomes

$$\mathcal{L}_D(\Lambda) = \sum_{i=1}^{N} \Lambda_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{k=1}^{N} \Lambda_i \Lambda_k y_i y_k K(x_i, x_k).$$

Next we will test the SVM approach on our mitral valve classification problem, using three separate kernels: using linear, polynomial and Gaussian. We have 30 normal and 28 regurgitation patients in our training set. The strain feature data provide 162 features for each one of these 58 patients. We save these data into a 58×162 matrix $XRaw$.

We compute the first two main principal components $v1$, $v2$ of the matrix $Xregion^T \cdot Xregion$, with the 58×9 matrix $Xregion$ on the region AP, which is one of 18 small regions. Projecting the data $Xregion$ onto the main components $v1, v2$, a smaller 58×2 matrix X_{AP} is created. Secondly, combining all 18 regions, we have the 58×36 matrix $Xpca = [X_{AP_1}, X_{AP_2}, \cdots, X_{AP_{18}}]$.

Then, the new matrix $Xpca$ is analyzed by PCA. In order to explain 95% of the

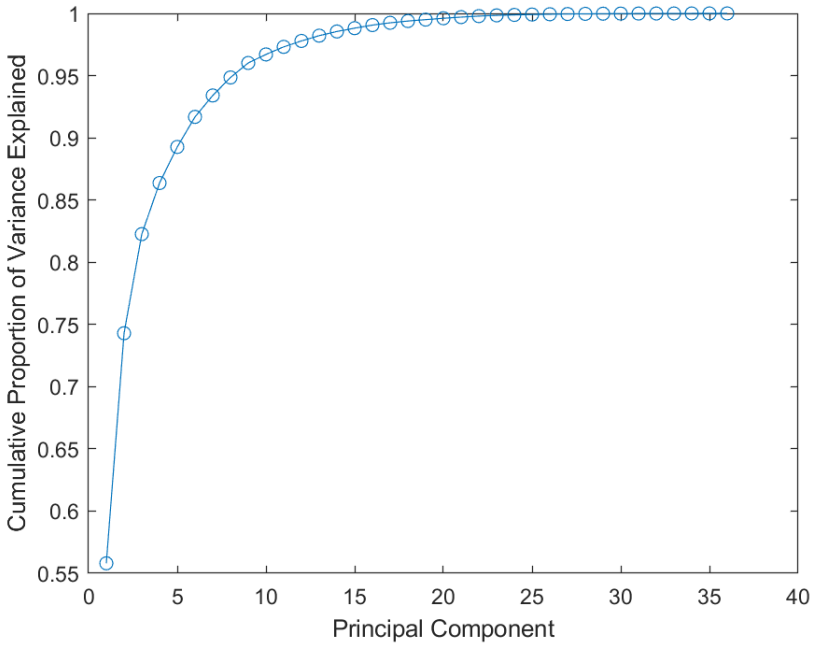


Figure 10.1: Total variance explained

variance, we choose the first 8 main components, which is illustrated in Figure 10.1. The new matrix $Xpca8$ will be a 58×8 matrix. Finally, we analyze the matrix $Xpca8$ using SVM relying on three kinds of kernel, linear kernel, polynomial kernel and Gaussian kernel.

Table 10.1: Support vector machines analysis (TN–true negatives, FP–false positives, FN–false negatives, TP–true positives).

Kernel Type	TN	FP	FN	TP	Accuracy
Linear	20	10	8	20	0.69
Polynomial of order 2	19	11	4	24	0.74
Polynomial of order 3	30	0	13	15	0.77
Polynomial of order 4	30	0	10	18	0.83
Gaussian	27	3	11	17	0.76

The SVM classifications based on polynomial kernels of degree 3 or 4 have roughly

80% accuracy. But the number of support vectors is larger than 30. We need to input more anatomic features into our SVM classification, such as, anterior leaflet area, posterior leaflet area, anterior coaptation line length, posterior coaptation line length, commissural diameter, tenting area, tenting height, etc. This will be tested in future work.

Bibliography

[1] R. Azencott, R. Glowinski, and A. M. Ramos, *A controllability approach to shape identification,* Applied Math. Letters, Vol. 21, No. 8, 2008, pp. 861-865.

[2] R. Azencott, R. Glowinski, J. W. He, A. Jajoo, Y. P. Li, A. Martynenko, R. H. W. Hoppe, S. Benzekry, S. H. Little, *Diffeomorphic matching and dynamic deformable surfaces in 3D medical imaging,* Computational Methods In Applied Mathematics, Vol. 10, No. 3, 2010, pp. 235-274.

[3] S. Arguillère, E. Trélat, A. Trouvé, L. Younes, *Registration of multiple shapes using constrained optimal control,* SIAM Journal Imaging Science, Vol. 9, No. 1, 2016, pp. 344-385.

[4] H. Bauschke, P. Combettes, *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*, Vol. 408, Springer, New York, 2011.

[5] M. D. Berg, O. Cheong, M. V. Kreveld, M. Overmars *Computational Geometry: Algorithms and Applications*, Springer, New York, 2008.

[6] C. D. Boor, *Spline Toolbox For Use With MATLAB*, MathWorks, 2005.

[7] S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein, *Distributed optimization and statistical learning via the alternating direction method of multipliers*, Found. Trends Mach. Learn., Vol. 3, No. 1, 2011, pp. 1-122.

[8] M. F. Beg, M. I. Miller, A. Trouvé, L. Younes, *Computing large deformations metrix mappings via geodesic flows of diffeomorphisms,* Int. J. Comp. Vol 61, No 2, 2005, pp. 139-157.

[9] V. Camion and L. Younes, *Geodesic interpolating splines,* International Workshop on Energy Minimization Methods in Computer Vision and Pattern Recognition. Vol. 2134, 2001, pp. 513-527.

[10] G. E. Christensen, *Deformable shape models for anatomy,* PhD Thesis, Department of Electrical Engineering, Sever Institute of Technology, Washington University, 1994

[11] G. E. Christensen, R. D. Rabbitt, and M. I. Miller, *Deformable templates using large deformation kinematics,* IEEE Transactions on Image Processing, Vol. 5, No. 10, 1996, pp. 1435-1447.

[12] P. Combettes and J. Pesquet, *Proximal splitting methods in signal processing,* in Fixed-Point Algorithms for Inverse Problems in Science and Engineering, Springer, New York, 2011, pp. 185-212.

[13] P. Combettes, *The convex feasibility problem in image recovery,* Advances in imaging and electron physics. Vol. 95. 1996, pp. 155-270.

[14] G. Dahlquist, A. Bjorck, *Numerical Methods in Scientific Computing: Volume 1*, Vol. 103. Siam, 2008.

[15] B. O′Donoghue, G. Stathopoulos, S. Boyd, *A Splitting Method for Optimal Control,* IEEE Transactions on Control Systems Technology, Vol. 21, No. 6, Nov. 2013, pp. 2432-2442.

[16] P. Dupuis, U. Grenander and M. Miller, *Variational problems on flows of diffeomorphisms for image matching*, Quarterly of Applied Math, 1998, pp. 587-600.

[17] J. Eckstein and D. Bertsekas, *On the Douglas-Rachford splitting method and the proximal point algorithm for maximal monotone operators,* Mathematical Programming, Vol. 55, No. 1, 1992, pp. 293-318.

[18] J. Freeman, *Combing diffeomorphic matching with image sequence intensity registration,* PhD Thesis, Department of Mathematics, University of Houston, 2014.

[19] J. Glaunes, A. Trouvé and L. Younes, *Diffeomorphic matching of distributions: A new approach for unlabelled point-sets and sub-manifolds matching,* Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004. Vol. 2. IEEE, 2004.

[20] U. Grenander, M. I. Miller, *Computational anatomy: an emerging discipline,* Quarterly of applied mathematics, Vol. 56, No. 4, 1998, pp. 617-694.

[21] H. Guo, A. Rangarajan and S. Joshi, *Diffeomorphic point matching,* Handbook of Mathematical Models in Computer Vision, Springer, Boston, 2006, pp. 205-219.

[22] R. Glowinski and J. L. Lions, *Exact and approximate controllability for distributed parameter systems,* Acta Numerica, 1995, pp. 159-328.

[23] D. Goldfarb and S. Ma, *Fast Multiple Splitting Algorithms for Convex Optimization,* SIAM Journal on Optimization, Vol. 22, No. 2, 2012, pp. 533-556.

[24] J. Glaunes, A. Q. Qiu, M. I. Miller, L. Younes, *Large deformation diffeomorphic metric curve mapping,* International Journal of Computer Vision, Vol. 80, 2008, pp. 317-336.

[25] T. Hastie, R. Tibshirani, J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, New York: Springer series in statistics, Vol, 1, No. 10, 2001.

[26] P. A. Helm, L. Younes, M. F. Beg, D. B. Ennis, C. Leclercq, O. P. Faris, E. McVeigh, D. Kass, M. I. Miller, R. L. Winslow, *Evidence of structural remodeling in the dyssynchronous failing heart,* Circulation Research, Vol. 98, 2006, pp. 125-132.

[27] S. Longo, E. Kerrigan, K. Ling, and G. Constantinides, *Parallel move blocking model predictive control,* in Proc. IEEE Decision Control Eur. Control Conf., 2011, pp. 1239-1244.

[28] S. Joshi, E. Klassen, A. Srivastava and I. Jermyn, *A novel representation for Riemannian analysis of elastic curves in r^n,* In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2007, pp. 1-7.

[29] I. T. Jolliffe, *Principal Component Analysis*, Springer Berlin Heidelberg, 2011.

[30] A. Jajoo, *Diffeomorphic matching and dynamic deformable shapes,* PHD Thesis, Department of Mathematics, University of Houston, 2011.

[31] S. Joshi and M. Miller, *Landmark matching via large deformation diffeomorphisms,* IEEE Transactions in Image Processing, Vol. 9, 2000, pp. 1357-1370.

[32] S. Joshi, *Large deformation diffeomorphisms and Gaussian random fields for statistical characterization of brain sub-manifolds,* PhD Thesis, Department of Electrical Engineering, Sever Institute of Technology, Washington University, 1998.

[33] K. C. E. Tallawi, P. Zhang, R. Azencott, J. W. He, E. Herrera, M. C. Pasha, J. Jacob, G. M. Lawrie, and W. A. Zoghbi, *Abstract 12634: Characterization of Mitral Valve Leaflet Deformation in Normal and Regurgitant Valves,* Circulation, 138, 2018, pp. A12634-A12634.

[34] M. I. Miller, A. Trouvé, and L. Younes, *On the metrics and Euler-Lagrange equations of computational anatomy,* Annual review of biomedical engineering, Vol. 4, No. 1, 2002, pp. 375-405.

[35] M. I. Miller, G. E. Christensen, Y. Amit, and U. Grenander, *Mathematical textbook of deformable neuroanatomies,* Proceedings of the National Academy of Science, Vol. 90, No. 24, 1993, pp. 11944-11948.

[36] D. L. Pham, C. Xu, and J .L. Prince, *Current methods in medical image segmentation,* Annual Review of Biomedical Engineering, Vol. 2, 2000, pp. 315-337.

[37] A. Quarteroni, R. Sacco, F. Saleri, *Numerical Mathematics,* Vol. 37, Springer Science & Business Media, 2010.

[38] R. T. Rockafellar and R. J. B. Wets, *Variational Analysis,* Vol. 317. Springer Science & Business Media, 2009.

[39] J. Shawe-Taylor and N. Cristianini,*Kernel Methods for Pattern Analysis,* Cambridge University Press, 2004.

[40] A. Trouvé, *An infinite dimensional group approach for physics based models in patterns recognition,* Technical Report, John Hopkins University, 1995.

[41] C. Twinnings, S. Marsland, and C. Taylor, *Measuring geodesic distances on the space of bounded diffeomorphisms,* BMVC. Vol. 2. 2002.

[42] A. Trouvé, *Diffeomorphisms Groups and Pattern Matching in Image Analysis,* International journal of computer vision, Vol. 28, 1998, pp. 213-221.

[43] P. A. , Van de Elsen, E. D. Pol, and M. A. Viergever, *Medical image matching-a review with classification,* IEEE Engineeing in Medicine and Biology, Vol. 12, 1993, pp. 26-39.

[44] L. Vandenberghe, *Lecture on proximal gradient method,* [Online] Available: http://www.seas.ucla.edu/˜vandenbe/236C/lectures/proxgrad.pdf, University of Carlifornia, 2019.

[45] Y. Wang and S. Boyd, *Fast model predictive control using online optimization,* IEEE Transactions on control systems technology, Vol. 18, No. 2, 2009, pp. 267-278.

[46] L. Younes, *Shapes and diffeomorphisms,* Springer. Berlin, Vol. 171, 2010.

[47] Q. Yue, *A Second Order Variational Approach For Diffeomorphic Matching of 3D Surfaces,* PhD Thesis, Department of Mathematics, University of Houston, 2013

[48] S. B. Zekry, G. Lawrie, S. Little, W. Zoghbi, J. Freeman, A. Jajoo, S. Jain, J. He, A. Martynenko, R. Azencott, *Comparative evaluation of mitral valve strain by deformation tracking in 3D-echocardiography,* Cardiovascular Engineering and Technology, Vol.3, No. 4, 2012, pp. 402-412.

[49] S. B. Zekry, J. Freeman, A. Jajoo, J. He, S. H. Little, G. M. Lawrie, R. Azencott and W. A. Zoghbi, *Patient-specific quantitation of mitral valve strain by computer analysis of three-dimensional echocardiography: a pilot study*, Circulation: Cardiovascular Imaging, Vol. 9, 2016, p. e003254.

[50] S. B. Zekry, J. Freeman, A. Jajoo, J. He, S. H. Little, G. M. Lawrie, R. Azencott and W. A. Zoghbi, *Effect of mitral valve repair on mitral valve leaflets Strain,* JACC, Cardiovascular Imaging, Vol. 11, No. 5, 2017, pp. 776-777.

[51] C. Cortes, and V. Vapnik. *Support-vector networks,* Machine Learning, Vol. 20, No.3, 1995, pp. 273-297.

[52] P. Wolfe, *A duality theorem for non-linear programming,* Quarterly of Applied Mathematics, Vol. 19, No. 3, 1961, pp. 239-244.

[53] B. Peter, *Lecture note: reproducing kernel Hilbert spaces,* [Online] Available: https://people.eecs.berkeley.edu/˜bartlett/courses/281b-sp08/7.pdf, University of Berkeley, 2008.