Distribution Agreement

In presenting this thesis as a partial fulfillment of the requirements for a degree from Emory University, I hereby grant to Emory University and its agents the non-exclusive license to archive, make accessible, and display my thesis in whole or in part in all forms of media, now or hereafter now, including display on the World Wide Web. I understand that I may select some access restrictions as part of the online submission of this thesis. I retain all ownership rights to the copyright of the thesis. I also retain the right to use in future works (such as articles or books) all or part of this thesis.

Claire Yilin Lin April 07, 2016

Line-to-Point Registration with Applications in Geometric Reconstruction of Coronary Stents

By

Claire Y. Lin

Lars Ruthotto
Advisor

Department of Mathematics and Computer Science

Lars Ruthotto
Advisor

Alessandro Veneziani
Co-advisor

James Nagy
Committee Member

Marina Piccinelli
Committee Member

2016

Line-to-Point Registration with Applications in Geometric Reconstruction of Coronary Stents

By

Claire Y. Lin

Lars Ruthotto
Advisor

An abstract of
a thesis submitted to the Faculty of Emory College of Arts and Sciences
of Emory University in partial fulfillment
of the requirements of the degree of
Bachelors of Science with Honors

Department of Mathematics and Computer Science

2016

Abstract

Line-to-Point Registration with Applications in Geometric Reconstruction of Coronary Stents

By Claire Y. Lin

Registration is a process of geometrically transforming one object to correspond to the other. It can utilized to align images, surfaces, or point clouds. The Iterative Closest Point algorithm is widely used in registration to achieve reconstruction of geometric shapes, but has drawbacks such as non-differentiability.

This thesis introduces a line-to-point registration method, useful in imposing a skeletal structure, defined by nodes and edges, on a given set of points in 2D or 3D. This method computes the distance from a point cloud to a skeletal structure using projections, and uses rigid, affine, and nonparametric transformations for distance minimization, taking into account regularization on the nonparametric transformation. The proposed approach can be utilized in the registration of two geometric objects, where one has a known structural skeleton, and the other is a point set.

In this thesis, this method is used to achieve correspondence between the undeformed and deformed configurations of a coronary prosthesis, called a bioresorbable stent. The undeformed configuration is represented by a skeleton of the prosthesis based on the manufacturer's design, and the deformed configuration is represented by a set of points obtained from medical images. Registration is used to automatize the geometric reconstruction of implanted coronary stents in patient-specific cases, to allow Computational Fluid Dynamics (CFD) analysis in the clinical trials at Emory University.

Line-to-Point Registration with Applications in Geometric Reconstruction of Coronary Stents

By

Claire Y. Lin

Lars Ruthotto
Advisor

A thesis submitted to the Faculty of Emory College of Arts and Sciences
of Emory University in partial fulfillment
of the requirements of the degree of
Bachelors of Science with Honors

Department of Mathematics and Computer Science

2016

Acknowledgements

I thank the Department of Mathematics and Computer Science at Emory University, for its great resources and its professional environment.

This thesis would not have been possible without the contributions of many. Hereby I wholeheartedly thank

Lars Ruthotto for his guidance and expertise in registration problems,

Alessandro Veneziani for his inputs and suggestions for our application,

Boyi Yang for his experience and the experimental data obtained,

Jan Modersitzki for his *Flexible Algorithm for Image Registration* toolbox,

Marina Piccinelli for her advise and commitment to the project,

Jim Nagy for his corrections and supervision on the thesis.

Contents

List of Figures

List of Examples

1 Introduction

Given two objects (e.g. points, lines, surfaces), the goal of registration is to find a reasonable transformation ϕ, such that one object, after the transformation, is similar to the other. Following the standard terminology in registration, we call the first object the template, and the other the reference.

For example, registration can be performed on two point clouds [7], two surfaces [14], or two images [12]. Here, we consider a case where the template is a set of line segments, and the reference is a set of points. This is useful in our application, as we discuss in Section 1.1.

To impose a structure, or shape, on a set of data points, registration techniques can be used to achieve correspondence. In this thesis, we explore an approach that matches the data points with a predefined set of line segments. In particular, we aim at deforming an object defined by nodes and edges, such that its distance to the point cloud is minimized. A simple illustration is shown in Figure 1.1.

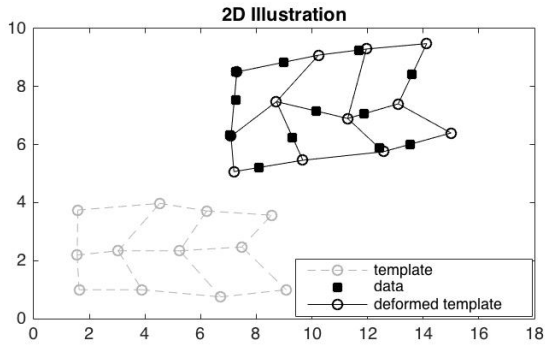

Figure 1.1: Imposing a pre-defined structure on a set of points.

1.1 Motivation

The motivation for this thesis is an application in the geometric reconstruction of a bioresorbable coronary stent, a device commonly used in cardiology to prevent the narrowing of coronary arteries. From an Optical Coherence Tomography (OCT) image of a patient undergoing Percutaneous Coronary Intervention (PCI) [16], one can obtain information of a cross-section of the coronary arteries, together with the struts of an implanted (hence deformed) stent. After the detection of stent structs from a set of OCT images, we can represent them by a set of points in 3D, each point corresponding to a strut of a deformed stent structure.

To enable further medical analysis (e.g. fluid dynamics simulations [15]) of the effects of a coronary stent on a patient, it is neccesary to reconstruct geometrically a deformed stent structure, based on the given 3D point cloud. Using registration, our goal is thus to find a reasonable deformation of a coronary stent that fits the observed data point cloud.

An illustration of a template and a reference in our application is shown in Figure 1.2. The template is defined using an undeformed stent structure based on the manufacturer's design, and here we refer to a horizontal circular frame as a "ring", and a vertical frame in between two rings as a "beam".

The process of stent reconstruction is previously accomplished by manually detecting and categorizing the observed data, as we discuss in Section 1.2. This can be inefficient due to manual interventions; hence the motivation of this thesis is to automatize the reconstruction process by registering an undeformed stent structure to a set of data points. We explore a method that accomplishes this task in either 2D or 3D.

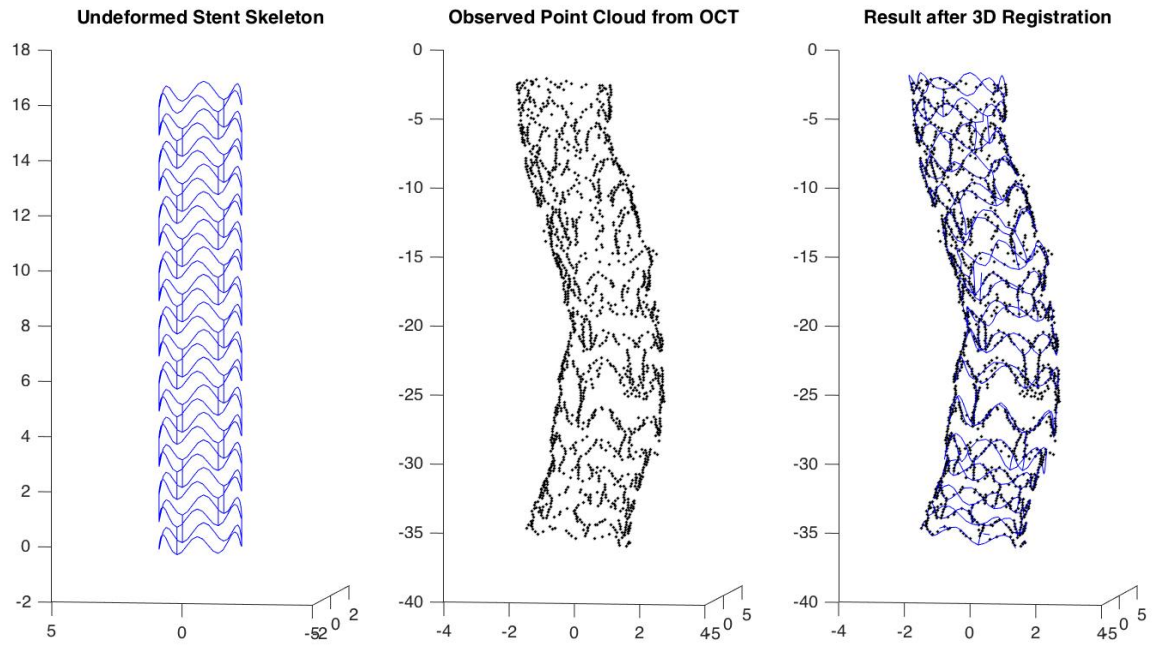

Figure 1.2: A pre-defined template with nodes $\mathbb{X}$, a reference point cloud $\mathbb{P}$, and a deformed template imposed on the point cloud.

1.2 Related Work

Method-wise, the registration of 2D or 3D shapes has been explored by many [6]. In particular, the Iterative Closest Point (ICP) is a widely-used algorithm for the reconstruction of 2D and 3D geometric entities.

The ICP method, first introduced in [7], can be used in registrations of point sets, line segment sets, curves, or surfaces [3]. Given a geometric object as a template, one can decompose it into a point set, and use the ICP algorithm to minimize its distance to a reference object [3]. Some proposed distance metrics used in ICP are the Euclidean distance between corresponding points [3], or the distance between a point and a tangent plane approximation of the surface at its corresponding point [7]. ICP is widely used in computer vision, pattern recognition and image analysis [8].

We discuss the ICP method, and its drawbacks (e.g. differentiability), in further

detail in Section 2, within the context of our application.

Application-wise, one approach explored by B. Yang [15] is to choose a particular section of the set of 3D data, and obtain a set of points in 2D by "unrolling" the 3D data. This enables us to manually connect and categorize the points in 2D, based on our knowledge of the stent structure. Combination of all sections results in an unrolled point cloud with categorization. Afterwards, a map back to 3D provides a categorized point cloud, and cubic spline is used to reconstruct the deformed stent structure in the patient's coronary arteries.

1.3 Contributions

In the theoretical aspect, we introduce a line-to-point registration method, useful for imposing a structure, defined by line segments, on a given set of points. Specifically, we use projections of data points onto line segments, by which differentiability is achieved, to compute the distance between the reference and the template. To minimize the computed distance, we propose a three-step transformation procedure, taking into account regularization.

From the application viewpoint, we use our method to automatize, and thus expedite, a critical step in the geometric reconstruction of implanted coronary stents. Given a 3D point cloud representing stent struts, we are able to achieve correspondence between a stent skeleton and the point cloud, thereby obtaining a deformed stent structure, useful for further analysis of its impact in the coronary arteries of a patient.

1.4 Outline

In Section 2, we consider the possibility of using the Iterative Closest Point algorithm for our applicational problem, and discuss its drawbacks in comparison with our line-to-point method. We then introduce our method in the context of our application in Section 3. We follow up with some numerical experiments in Section 4, and provide directions for future work in Section 5.

2 Iterative Closest Point

In this section, we discuss the ICP as a method alternative to line-to-point, that one could use to achieve our goal of imposing a stent structure on a set of observed data. We then discuss the disadvantages of the ICP method, in comparison with the line-to-point registration approach.

2.1 The ICP Algorithm

To apply the ICP to the reconstruction of a coronary stent, consider a point cloud of nodes on an undeformed stent, $\mathbb{X} = \{x_1, x_2, \ldots, x_{N_x}\} \subset \mathbb{R}^d$ ($d = 2$ or 3), and a point cloud of observed data $\mathbb{P} = \{p_1, p_2, \ldots, p_{N_p}\} \subset \mathbb{R}^d$. One way to find correspondence is to use the ICP algorithm to minimize the distance between the two point clouds $\mathbb{X}$ and $\mathbb{P}$.

Briefly speaking, the ICP algorithm starts with the identity transformation $\phi^{(0)}$, and iteratively finds a transformation $\phi^{(k)}$ until it reaches some stopping criterion (e.g. number of iterations, change in misfit, etc.). In the kth iteration ($k > 0$), we minimize the distance between the transformed template $\phi^{(k)}(\mathbb{X}) = \{\phi^{(k)}(x_1), \phi^{(k)}(x_2), \ldots, \phi^{(k)}(x_{N_x})\} = \{\phi_1^{(k)}, \phi_2^{(k)}, \ldots, \phi_{N_x}^{(k)}\}$ and the reference $\mathbb{P}$. This distance can be written as

$$D\big(\phi^{(k)}(\mathbb{X}), \mathbb{P}\big) = \sum_{i=1}^{N_x} \|\phi_i^{(k)} - \widetilde{p}_i\|_2^2, \quad \text{where } \widetilde{p}_i = \operatorname*{argmin}_{p \in \mathbb{P}} \|\phi_i^{(k-1)} - p\|_2^2. \tag{1}$$

Here we use a discrete representation for distance D, as we consider the case when ϕ is discretized on the point cloud $\mathbb{X}$. In general, however, ϕ can be continuous.

Using ICP, we can compute the rigid registration of two point sets using closed-form methods, such as singular value decomposition, by expressing the transformation with a rotation matrix and a translation vector [1]. One can also introduce a scale matrix,

and achieve registration with scaling [8].

2.2 ICP vs. Line-to-Point

Here we discuss the differentiability, and visualization, as two major advantages the line-to-point approach has, in our application, over the ICP method.

Due to the pointwise nature of the template in the ICP method, the problem is discrete and therefore non-differentiable. To see this in more detail, let us look at a situation with ICP in Example 2.1, and compare it with a similar case using the line-to-point approach in Example 2.2.

Example 2.1: Non-differentiability of ICP

Consider the case when in the first iteration, a point $x_i \in \mathbb{X}$ has equal distances to two points $p_1, p_2 \in \mathbb{P}$ (Figure 2.1). The discrete nature of ICP causes an ambiguity of choosing between p_1 and p_2 for the value of $\widetilde{p}_i$ in (1), making it impossible to formulate the expression of $\nabla D\big(\phi(\mathbb{X}), \mathbb{P}\big)$ in a unique manner.

To be more specific, since p_1 and p_2 give us the same value for the distance, choosing either one does not affect the overall $D\big(\phi(\mathbb{X}), \mathbb{P}\big)$. However, if p_1 is chosen, and ϕ causes x_i to move further from p_1 and closer to p_2 (Figure 2.1), then the Taylor series, using p_1 in the expression for $\nabla D\big(\phi(\mathbb{X}), \mathbb{P}\big)$, predicts the objective function to increase in value. This is not true if p_2 is chosen, since then the gradient would predict a decrease in the overall distance, which is what we desire.

Figure 2.1: The pointwise nature of ICP causes its non-differentiability.

Example 2.2: Differentiability of Line-to-point Approach

In our line-to-point registration approach, however, we use the projection of each $p_i \in \mathbb{P}$ onto the edges in our template. This requires a template with pre-defined line segments, and in our application, it comes from our knowledge of the skeletal structure of an undeformed stent.

We discuss projections in more detail in Section 3.2.1. As a brief example, let $\widehat{p_i}$ be the projection of p_i onto the edge defined by nodes x_1, x_2 (Figure 2.2), then the problem is continuous and differentiable up to the length of the edge, thus relieving the non-differentiability of ICP. When x_1 is moved further from p_i (Figure 2.2), projection onto the edge defined by ϕ_1 and ϕ_2 allows differentiability on this edge.

The line-to-point method is applicable to the case of a coronary stent, since we do not require any $p_i \in \mathbb{P}$ to land on a particular $x_j \in \mathbb{X}$, but only to get close enough to our pre-defined template structure. We present the results in our application using this approach in Section 4.1.

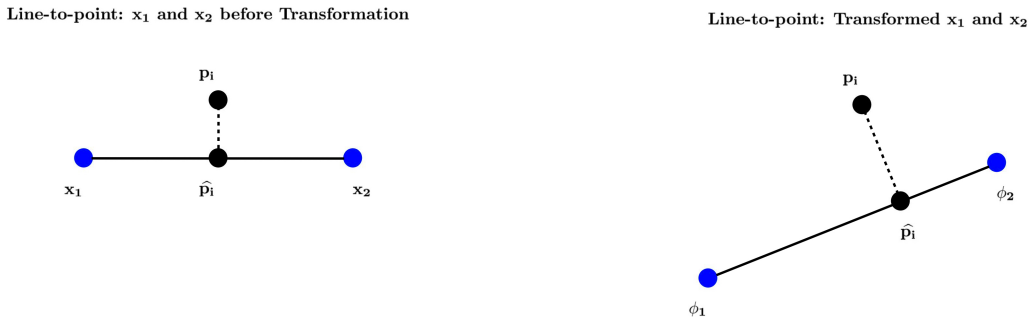

Figure 2.2: Projection in line-to-point registration makes the problem differentiable.

Another advantage the line-to-point approach has over ICP is its ability to be visualized and deformed directly.

In (1), the step of finding the closest point for each x_i in $\mathbb{P}$ can be problematic in practice. This is because we only have the knowledge of $\mathbb{X}$ as a set of points, and do not have control over how $\mathbb{X}$ as a whole is moved by ϕ. In our application of a coronary stent, however, $\mathbb{X}$ is more than a disordered point cloud; each point in $\mathbb{X}$ is connected to some other points by the structure of a coronary stent. Therefore, using merely the ICP for nonlinear transformations, we can end up with an unreasonable stent structure (e.g. overlapping of stent).

One way to resolve this issue is to impose a grid (i.e. a partitioning of the region of interest into a number of congruent cells [12]) on $\mathbb{X}$, on which we interpolate a nonlinear transformation ϕ with respect to this grid [12], so that we have control over the movement of $\mathbb{X}$. In the end, the grid transformed by ϕ can also help visualize the reasonability of our resulting $\phi(\mathbb{X})$.

Using our line-to-point approach, however, such a grid is not necessary. Rather, the transformed template itself gives a nice visualization of the reasonability of our solution.

Further detail on our template and the general approach is discussed in Section 3.

3 Line-to-Point Registration

In this section, we introduce the procedure of our line-to-point registration approach. We split the section into subsections based on discussions of a template, the objective function, and numerical optimization.

At each subsection, we review some theoretical concepts if necessary. We then discuss the approach in the context of our application (i.e. reconstruction of a coronary stent), but the method can be extended to other applications. Specifically, if we have information of a data set, and a desired skeletal structure to act as a template (Section 3.1), we can deform the structure to fit the given data.

3.1 Template

Given a set of data points, our template is a piecewise linear object defined by nodes and edges. Our objective is to deform the template such that its distance to the point cloud is minimized.

As in our notation in Section 2.1, let $\mathbb{X} = \{x_1, x_2, \ldots, x_{N_x}\} \subset \mathbb{R}^d$ be a point cloud of nodes in our template, and $\mathbb{P} = \{p_1, p_2, \ldots, p_{N_p}\} \subset \mathbb{R}^d$ be a point cloud of observed data. In addition, we define edges in our template by $\mathbb{E} = \big\{\{i, j\} : \exists \text{ a line segment between } x_i \text{ and } x_j\big\}$.

From now on, we represent our template as $G(\mathbb{X}, \mathbb{E})$. To relate this notation to the traditional representation of an undirected graph $G(V, E)$ [4], simply define a vertex function $f : V \to \mathbb{R}^d$ by $f(i) = x_i$, and let $\mathbb{E} = E$.

In our application, the template is an undeformed stent skeleton, which is built based on the design of a bioresorbable coronary stent [15]. This can be done in both 2D and 3D (Figure 3.1). Transformation of the template is done by directly transforming the nodes $\mathbb{X}$ (Section 3.2.2). Since the edges $\mathbb{E}$ are defined in terms of the indices of

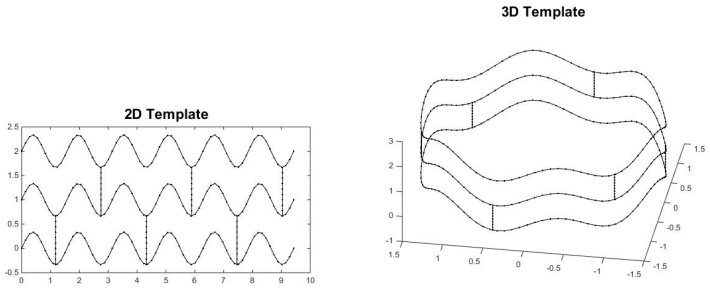

Figure 3.1: Sections of 2D and 3D templates in our application.

nodes in the template, movement of the entire skeleton is an immediate consequence of the movement of $\mathbb{X}$. In the entire process, $\mathbb{E}$ remains unchanged.

3.2 Objective Function

Our registration problem is to find a reasonable transformation ϕ of a template $G(\mathbb{X}, \mathbb{E})$, such that its distance to the reference $\mathbb{P}$ is minimized. To this end, the objective function $J : \mathbb{R}^{d \cdot N_x} \to \mathbb{R}$ can be written as

$$J\big(\phi(\mathbb{X})\big) = D\Big(G\big(\phi(\mathbb{X}), \mathbb{E}\big), \mathbb{P}\Big) + \sum_i \alpha_i S_i\Big(G\big(\phi(\mathbb{X}), \mathbb{E}\big)\Big)\,, \tag{2}$$

where D denotes a distance measure (Section 3.2.3), α_i is a regularization parameter, and S_i, as a regularizer, measures the reasonability of a transformation (Section 3.2.4). Depending on whether the transformation is parametric (i.e. whether the transformation can be written as a linear combination of basis functions, with coefficients as elements in a parameter $\mathbf{w}$ [12]), we minimize J with respect to either the parameter $\mathbf{w}$, or the transformed nodes $\phi(\mathbb{X}) = \{\phi(x_1), \phi(x_2), \ldots, \phi(x_{N_x})\} = \{\phi_1, \phi_2, \ldots, \phi_{N_x}\}$ (Section 3.2.2).

In this subsection, we present the objective function by discussing projection, parametric and nonparametric transformations, distance measure, and regularization.

3.2.1 Projection

The projection of a point p on a line segment is a point $\widehat{p}$ that has the shortest distance to p. Since a line segment is a closed and convex set in $\mathbb{R}^d$, the projection is unique [5]. In our problem, we want to find for each $p \in \mathbb{P}$ the closest projection onto all the edges in $G(\mathbb{X}, \mathbb{E})$.

Consider a point $p_k \in \mathbb{P}$, and its projection on an edge in $G(\mathbb{X}, \mathbb{E})$ with endpoints $x_i, x_j \in \mathbb{X}$,

$$\widehat{p}_k^{(i,j)} = x_i + t_k^{(i,j)}(x_j - x_i)\,, \tag{3}$$

$$\text{where } t_k^{(i,j)} = \begin{cases} 0 & \text{if } c_k^{(i,j)} < 0, \\ c_k^{(i,j)} & \text{if } 0 \le c_k^{(i,j)} \le 1, \\ 1 & \text{if } c_k^{(i,j)} > 1, \end{cases} \quad \text{and } c_k^{(i,j)} = \frac{(x_j - x_i)^T(p_k - x_i)}{(x_j - x_i)^T(x_j - x_i)}\,.$$

In the expression of $t_k^{(i,j)}$ we normalize $(x_j - x_i)$, and $t_k^{(i,j)}$ takes care of projection $\widehat{p}_k^{(i,j)}$ both in between and at the endpoints of the line segment [5]. The distance from point p_k to the line segment $\{x_i, x_j\}$ can then be written as $\|\widehat{p}_k^{(i,j)} - p_k\|_2^2$.

Using the projection defined in (3), we perform projections of p_k onto all edges in our template $G(\mathbb{X}, \mathbb{E})$, and compute the distance from p_k to each edge. We can then pick the minimum of these distances, and denote the corresponding projection as $\widehat{p}_k$, and the scalar as t_k. We also record the corresponding edge index. Repeat this process for all $p \in \mathbb{P}$, and this is called the brute-force search.

For future work, the above step can be improved by using more efficient search techniques. For example, one can pre-compute the region in $\mathbb{R}^d$ closest to a given $p_i \in \mathbb{P}$, using the Delaunay triangulation, or the Voronoi diagram [2]. We can then project p_i onto the edges corresponding to the nodes in $\mathbb{X}$ in this region, thus reducing the number of searches needed.

Let us look at how to implement projection numerically. We can represent the

nodes $\mathbb{X}$ as a vector of length $d \cdot N_x$, denoted by $\mathbf{X}$. Then $\forall k = 1, 2, \ldots, N_p$, write the corresponding projection with minimal distance $\widehat{p}_k\big(G(\mathbb{X}, \mathbb{E})\big)$ as

$$\widehat{p}_k(\mathbf{X}) = Q_k^{(1)}\mathbf{X} + t_k \cdot (Q_k^{(2)} - Q_k^{(1)})\mathbf{X},$$

$$\text{where } t_k = \begin{cases} 0 & \text{if } c_k < 0, \\ c_k & \text{if } 0 \le c_k \le 1, \\ 1 & \text{if } c_k > 1, \end{cases} \quad \text{and } c_k = \frac{\big((Q_k^{(2)} - Q_k^{(1)})\mathbf{X}\big)^T \big(p_k - Q_k^{(1)}\mathbf{X}\big)}{\|(Q_k^{(2)} - Q_k^{(1)})\mathbf{X}\|_2^2},$$

and $Q_k^{(1)}$, $Q_k^{(2)}$ are matrices of sizes d by $d \cdot N_x$, obtained from the recorded edge indices, such that $Q_k^{(1)}\mathbf{X}$ and $Q_k^{(2)}\mathbf{X}$ are the endpoints of the closest edge corresponding to p_k.

Using the above, we repeat the process N_p times, to obatin a point cloud of projections $\widehat{\mathbb{P}}\big(G(\mathbb{X}, \mathbb{E})\big) = \big\{\widehat{p}_1\big(G(\mathbb{X}, \mathbb{E})\big), \widehat{p}_2\big(G(\mathbb{X}, \mathbb{E})\big), \ldots, \widehat{p}_{N_p}\big(G(\mathbb{X}, \mathbb{E})\big)\big\}$. Observe that each $\widehat{p}_k : \mathbb{R}^{d \cdot N_x} \to \mathbb{R}^d$, when written as a function of $\mathbf{X}$, is differentiable except at the endpoints of the corresponding line segment. This differentiability, as mentioned before in Section 2.2, is an advantage in our application over the non-differentiable ICP method.

To get the gradient of $\widehat{p}_k$ with respect to $\mathbf{X}$, first find the gradient ∇t_k using the quotient rule, then find the Jacobian $\mathbf{J}\widehat{p}_k$ using the product rule. To represent it explicitly, let $Q_k = Q_k^{(2)} - Q_k^{(1)}$, then

$$\nabla t_k = \begin{cases} 0 & \text{if } t_k = 0 \text{ or } t_k = 1, \\ \frac{\|Q_k\mathbf{X}\|_2^2 (Q_k^T p_k - Q_k^T Q_k^{(1)}\mathbf{X} - Q_k^{(1)T} Q_k \mathbf{X}) - 2\mathbf{X}^T Q_k^T (p_k - Q_k^{(1)}\mathbf{X}) Q^T Q \mathbf{X}}{\|Q_k\mathbf{X}\|_2^4} & \text{otherwise,} \end{cases}$$

$$\text{and} \quad \mathbf{J}\widehat{p}_k = Q_k^{(1)} + t_k Q_k + Q_k \mathbf{X} \nabla t_k^T.$$

We can perform this differentiation procedure on all the points in $\widehat{\mathbb{P}}\big(G(\mathbb{X}, \mathbb{E})\big)$. In the end, if we write the point cloud $\mathbb{P}$ and the projection point cloud $\widehat{\mathbb{P}}$ as vectors of

length $d \cdot N_p$, denoted respectively by $\mathbf{P}$ and $\widehat{\mathbf{P}}$, then we can represent $\widehat{\mathbf{P}} : \mathbb{R}^{d \cdot N_x} \to \mathbb{R}^{d \cdot N_p}$ as a function of $\mathbf{X}$.

From the application viewpoint, we have found, for each point in the observed data $\mathbb{P}$, a corresponding projection in our undeformed stent template $G(\mathbb{X}, \mathbb{E})$. This projection, as we discuss in Section 3.2.3, is used to define the distance measure D in (2).

3.2.2 Transformations

In this thesis, we consider both parametric and nonparametric transformations in our objective function. Here we review the concepts of rigid and affine transformations, and discuss transformations in the context of our application.

Consider a node $x_i \in \mathbb{X}$, and its image ϕ_i under a transformation ϕ. In the case of a parametric transformation, let $\mathbf{w} = [w_1 \ w_2 \ \ldots]^T$ denote the transformation parameter.

Rigid transformation allows translations and rotations. When $d = 2$, $\mathbf{w} \in \mathbb{R}^3$, and a rigid transformation can be written in the form

$$\phi_i = \begin{bmatrix} \cos(w_1) & -\sin(w_1) \\ \sin(w_1) & \cos(w_1) \end{bmatrix} x_i + \begin{bmatrix} w_2 \\ w_3 \end{bmatrix}.$$

When $d = 3$, we have $\mathbf{w} \in \mathbb{R}^6$, and a linear system with a product of 2D plane rotation matrices defined by w_1, w_2, and w_3 [12].

Affine linear transformation allows shearing and scaling. When $d = 2$, $\mathbf{w} \in \mathbb{R}^6$, and an affine linear transformation can be written in the form

$$\phi_i = \begin{bmatrix} w_1 & w_2 \\ w_4 & w_5 \end{bmatrix} x_i + \begin{bmatrix} w_3 \\ w_6 \end{bmatrix}.$$

When $d = 3$, we have $\mathbf{w} \in \mathbb{R}^{12}$ [12].

In the rigid and affine transformations above, all $\phi_i \in \phi(\mathbb{X})$ depend on $\mathbf{w}$, and thus the objective function in (2) is minimized with respect to $\mathbf{w}$. In cases when a transformation cannot be parameterized, the objective function is minimized with respect to the transformed point cloud $\phi(\mathbb{X})$ directly.

Example 3.1: Three-step Transformation in 2D

In practice, we follow a three-step transformation process to achieve registration. As an illustration, we use the example at the beginning of Section 1. As shown in Figure 3.2, the misfit $J = D\Big(G\big(\phi(\mathbb{X}), \mathbb{E}\big), \mathbb{P}\Big)$ decreases in each step. After nonparametric transformation, we observe visually that the transformed template $G\big(\phi(\mathbb{X}), \mathbb{E}\big)$ fits the data $\mathbb{P}$.

Here we make use of the `trafo` function, provided by FAIR [12], to perform rigid and affine linear transformations.

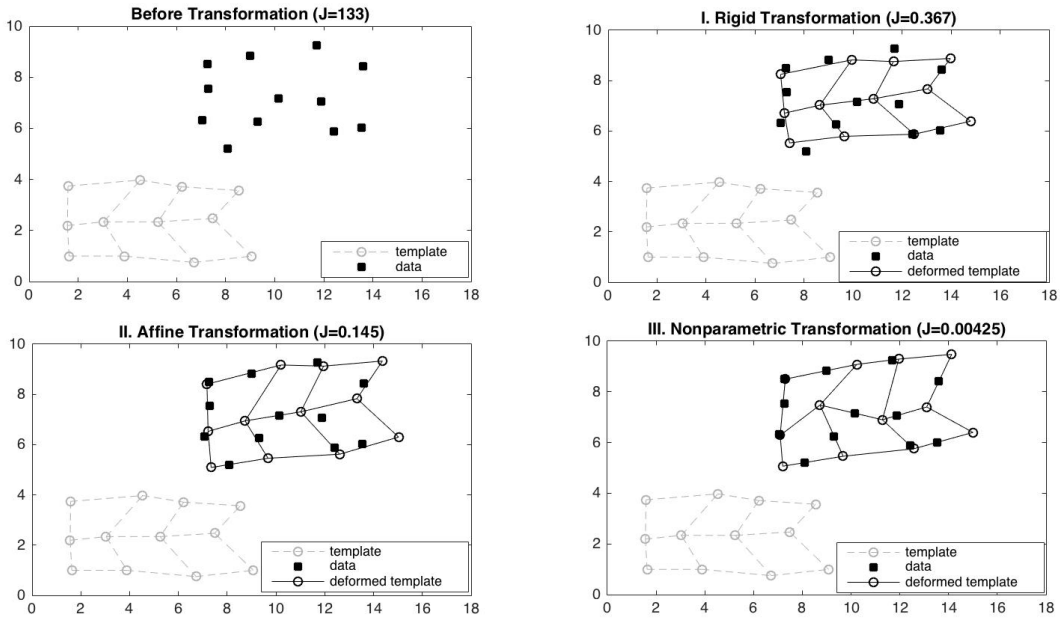

Figure 3.2: Deforming a pre-defined template to match a set of points, using three steps of transformations.

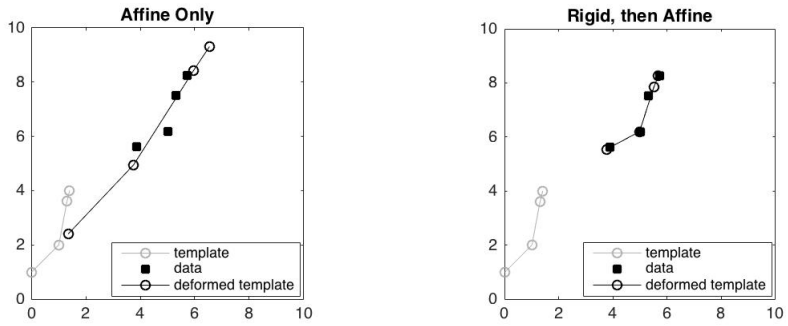

Figure 3.3: An example of performing affine transformation only, and invoking rigid transformation before affine transformation.

Example 3.2: Necessity of Rigid Transformation before Affine

Although rigid transformation can be seen as a particular affine transformation [12], note that it is necessary here to perform rigid transformation separately. This is because of the projection method we are using. As an example, we start with a template far away from the data point cloud, find the projections with minimal distances, and immediately perform affine linear transformation (Figure 3.3). In this case, the closest projections cover only a small portion of our template, and we get a result that does not preserve the lengths of edges in the template. This can be undesirable in many applications.

In the coronary stent case, the three-step transformation process can be visualized by the deformation of a stent structure $G(\mathbb{X}, \mathbb{E})$, so that its distance to the observed point cloud $\mathbb{P}$ gets minimized in each step.

With the knowledge of projection and transformations, we are now ready to define a distance measure D.

3.2.3 Distance Measure

At this stage, we have knowledge of a transformed template $G\big(\phi(\mathbb{X}), \mathbb{E}\big)$, and a reference point cloud $\mathbb{P}$. The idea is to obtain a set of closest projections $\widehat{\mathbb{P}}\Big(G\big(\phi(\mathbb{X}), \mathbb{E}\big)\Big)$, and compute the distance between the data and the projections. By this, we can write the distance as

$$D\Big(G\big(\phi(\mathbb{X}), \mathbb{E}\big), \mathbb{P}\Big) = \frac{1}{2}\sum_{i=1}^{N_p} D\Big(G\big(\phi(\mathbb{X}), \mathbb{E}\big), p_i\Big) = \frac{1}{2}\sum_{i=1}^{N_p} \Big\| p_i - \widehat{p}_i\Big(G\big(\phi(\mathbb{X}), \mathbb{E}\big)\Big)\Big\|_2^2 ,$$

where we define the distance from a point $p_i \in \mathbb{P}$ to a template $G(\mathbb{X}, \mathbb{E})$ to be the L_2-norm of the distance from p_i to its projection $\widehat{p_i}$ onto the template.

From the numerical viewpoint, since $\mathbb{E}$ and $\mathbb{P}$ never change throughout the process, $D : \mathbb{R}^{d \cdot N_x} \to \mathbb{R}$ can be written with respect to $\mathbf{X}$ using $\widehat{\mathbf{P}}$ defined in Section 3.2.1:

$$D(\mathbf{X}) = \frac{1}{2}\|\mathbf{P} - \widehat{\mathbf{P}}(\mathbf{X})\|_2^2 .$$

The gradient of D can be obtained by chain rule using the computed Jacobian $\mathbf{J}\,\widehat{\mathbf{P}}$:

$$\nabla D(\mathbf{X}) = \big(\mathbf{P} - \widehat{\mathbf{P}}(\mathbf{X})\big)\mathbf{J}\,\widehat{\mathbf{P}}(\mathbf{X}) ,$$

and the Hessian of D can be approximated by letting

$$\nabla^2 D(\mathbf{X}) \approx \mathbf{J}\,\widehat{\mathbf{P}}(\mathbf{X})^T \mathbf{J}\,\widehat{\mathbf{P}}(\mathbf{X}) .$$

Hence with transformation ϕ, the distance $D\Big(G\big(\phi(\mathbb{X}), \mathbb{E}\big), \mathbb{P}\Big)$ can be represented as a function of $\phi(\mathbf{X})$. This distance measure provides us a way to represent the misfit between our transformed template $G\big(\phi(\mathbb{X}), \mathbb{E}\big)$ and the reference $\mathbb{P}$.

We now proceed to look at the remaining term in the objective function (2), the

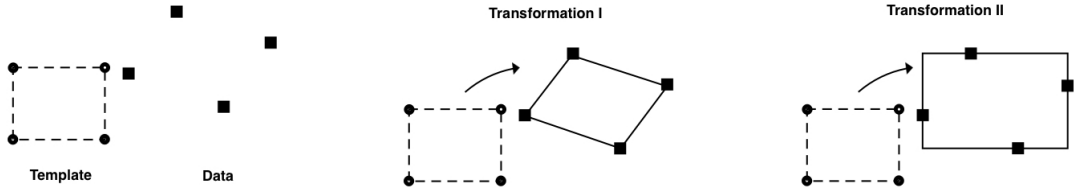

Figure 3.4: Both transformations of the template qualify as solutions.

regularizer.

3.2.4 Regularization

Registration problems, along with other inverse problems, are usually not well-posed in the sense of Hadamard. That is, solutions to these problems either do not exist, are not unique, or do not depend continuously on the data [9].

Example 3.3: Ill-posedness of the Problem

In our case, for example, the problem of minimizing the distance measure is ill-posed, since there is not a unique solution. As a simple illustration, let us consider the problem in Figure 3.4. Both transformations I and II are solutions, since both transformed templates end up fitting the data, minimizing $D\Big(G\big(\phi(\mathbb{X}),\mathbb{E}\big),\mathbb{P}\Big)$. However, they are visually different, as one roughly preserves the length of each edge in the template, and the other keeps the overall rectangular shape. Which one to choose would depend on our practical purposes, and to obtain a reasonable solution, regularization is necessary.

The purpose of a regularizer $S_i\Big(G\big(\phi(\mathbb{X}),\mathbb{E}\big)\Big)$ is to avoid undesirable results due to ill-posedness, by penalizing unrealistic transformations of $\mathbb{X}$. Regularization parameters, $\alpha_i > 0$, are used for compromising between similarity and reasonability [12].

In our application, for example, it is not enough to require the transformed template to fit the given data. Because of material properties of a coronary stent, a reasonable solution would restrict the length the skeletal stent from being unrealistically stretched, squeezed, or deformed in different directions.

Here we consider two regularizers for nonparametric transformation: a length regularizer S_L, and a direction regularizer S_D.

Our length regularizer S_L considers each edge in the transformed template $G\big(\phi(\mathbb{X}), \mathbb{E}\big)$, and penalize change in its length. Let $S_L : \mathbb{R}^{d \cdot N_x} \to \mathbb{R}$ be defined by

$$S_L\Big(G\big(\phi(\mathbb{X}), \mathbb{E}\big)\Big) = \frac{1}{2} \sum_{i=1}^{N_e} \left(\frac{\|\phi(x_{i,1}) - \phi(x_{i,2})\|_2^2}{\|x_{i,1}^{\text{tem}} - x_{i,2}^{\text{tem}}\|_2^2} - 1 \right)^2,$$

where N_e is the size of $\mathbb{E}$, and $x_{i,1}^{\text{tem}}$, $x_{i,2}^{\text{tem}}$ and $\phi(x_{i,1})$, $\phi(x_{i,2})$ denote the left and right endpoints of the ith edge in our template respectively before and after a nonparametric transformation ϕ.

Our direction regularizer S_D considers the directions toward which two nodes on an edge move, and penalize the distances moved in different directions. Let $S_D : \mathbb{R}^{d \cdot N_x} \to \mathbb{R}$ be defined by

$$S_D\Big(G\big(\phi(\mathbb{X}), \mathbb{E}\big)\Big) = \frac{1}{2} \sum_{i=1}^{N_e} \Big\| \big(\phi(x_{i,1}) - x_{i,1}^{\text{tem}}\big) - \big(\phi(x_{i,2}) - x_{i,2}^{\text{tem}}\big) \Big\|_2^2 .$$

This regularizer measures the variations of nodes in the template; its purpose is thus analogous to that of a diffusion operator [12].

Note that in our application, we define the template as a skeletal stent structure with equal-length edges, and no weight is required in this case. In general, if lengths of the edges differ greatly, we can introduce weights on the edges in the template (e.g. let $w_i = \|x_{i,1}^{\text{tem}} - x_{i,2}^{\text{tem}}\|_2^2$ for $i = 1, \ldots, N_e$). One can relate this template to a weighted

graph [4].

To implement S_L numerically as a function of $\mathbf{X}$, first define a matrix A_0 of size N_e by N_x in terms of the edges $\mathbb{E}$ in our template, such that $A(i,j) = 1$, where $x_j = x_{i,1}$, and $A(i,k) = -1$, where $x_k = x_{i,2}$. Then the Kronecker product [12] $A = I_{d\times d} \otimes A_0$, where $I_{d\times d}$ is the d by d identity matrix, takes the value of $[x_{1,1} - x_{1,2} \ \dots \ x_{N_e,1} - x_{N_e,2}]^T$ written as a vector of length $d \cdot N_e$. Here A has size $d \cdot N_e$ by $d \cdot N_x$.

Let $\mathbf{X}^{\text{tem}}$ denote the vector $\mathbf{X}$ prior to nonparametric transformation, then the only variable is $\mathbf{X}$, and the length regularizer can be written as

$$S_L(\mathbf{X}) = \frac{1}{2}\|r_L(\mathbf{X})\|_2^2, \quad \text{where } r_L(\mathbf{X}) = I \cdot (A\mathbf{X} \odot A\mathbf{X}) \oslash (A\mathbf{X}^{\text{tem}} \odot A\mathbf{X}^{\text{tem}}) - 1\,.$$

Here $\odot$ and $\oslash$ denote respectively element-wise multiplication and division, and $I = I_{1\times d} \otimes I_{N_e \times N_e}$ has size N_e by $d \cdot N_e$, where $I_{1\times d}$ is a 1 by d matrix with 1 as entries, and $I_{N_e\times N_e}$ is the N_e by N_e identity matrix.

Using this notation, the gradient can be expressed as

$$\nabla S_L(\mathbf{X}) = r_L(\mathbf{X})^T \nabla r_L(\mathbf{X}), \quad \text{where } \nabla r_L(\mathbf{X}) = 2 \cdot I \oslash (A\mathbf{X}^{\text{tem}} \odot A\mathbf{X}^{\text{tem}}) \cdot \text{diag}(A\mathbf{X}) \cdot A\,,$$

and $\text{diag}(A\mathbf{X})$ denotes a diagonal matrix with entries in $A\mathbf{X}$ on its main diagonal [10]. We can also approximate the Hessian by letting

$$\nabla^2 S_L(\mathbf{X}) \approx \nabla r_L(\mathbf{X})^T \nabla r_L(\mathbf{X})\,.$$

In a similar manner, we can express S_D using A defined above, such that

$$S_D(\mathbf{X}) = \frac{1}{2}\|r_D(\mathbf{X})\|_2^2, \quad \text{where } r_D(\mathbf{X}) = A(\mathbf{X} - \mathbf{X}^{\text{tem}})\,.$$

Here the gradient and Hessian are respectively

$$\nabla S_D(\mathbf{X}) = r_D(\mathbf{X})^T A\,, \quad \text{and} \quad \nabla^2 S_D(\mathbf{X}) = A^T A\,.$$

This Hessian of size $d \cdot N_x$ by $d \cdot N_x$ is exactly $I_{d\times d} \otimes L_G$, where L_G is the Laplacian [4] of an undirected graph $G(V, E)$, whose relation to our template $G(\mathbb{X}, \mathbb{E})$ is discussed before in Section 3.1.

With a nonparametric transformation ϕ, S_L and S_D is computed with respect to $\phi(\mathbf{X})$. Combining the numerical representations of the distance measure and the regularizers, we can now express the objective function (2) for nonparametric transformation as

$$J\big(\phi(\mathbf{X})\big) = D\big(\phi(\mathbf{X})\big) + \alpha_L S_L\big(\phi(\mathbf{X})\big) + \alpha_D S_D\big(\phi(\mathbf{X})\big)\,,$$

where $\alpha_L, \alpha_D > 0$ are regularization parameters of S_L and S_D. For parametric transformations, we use the objective function without regularization.

3.3 Numerical Optimization

In each of the three transformation steps, we minimize the objective function J by numerical optimization. Here we consider the Gauss-Newton method with the Armijo line search to solve the minimization problem [13].

Here we use the `GaussNewton` function, provided by FAIR [12], to perform numerical optimization.

4 Numerical Experiments

In our application, detection of stent struts has been performed by T. Han [11] and B. Yang [15], based on OCT images provided by the Division of Cardiology at Emory Healthcare. We thus start with a 3D point cloud, on which we proceed to test the line-to-point registration approach.

In this section, we first demonstrate the entire process in 3D by an example. We then discuss some supplementary steps as we implement the method in our practical application.

4.1 3D Case Step-by-Step Demonstration

We perform experiments in 3D, as it is more effective for our application. However, the procedure described here can be applied to a 2D point cloud, as Example 3.1 illustrates.

Here we use a section of obtained data for demonstration. For visualization purposes, let the reference $\mathbb{P}$ correspond to three rings in a stent, and define a template $G(\mathbb{X}, \mathbb{E})$ (Figure 4.1). In this case, $N_x = 162$, $N_e = 168$, and $N_p = 230$.

Since $G(\mathbb{X}, \mathbb{E})$ and $\mathbb{P}$ are far from each other, we first use rigid transformation to bring them together. This step can be replaced in practice, as we discuss in Section 4.2.2. For stopping criterion, here we let the maximal number of iterations be 15. In this case it takes 10 iterations, and the template is brought close to our data, with a misfit of $J \approx 28.7$ (Figure 4.2).

Here the -1th iteration refers to the state before registration, and the 0th iteration represents that before a transformation is performed.

We then use affine transformation to deform the template by shearing and scaling. Here we let the maximal number of iterations be 20, and it takes 15 iterations, with a

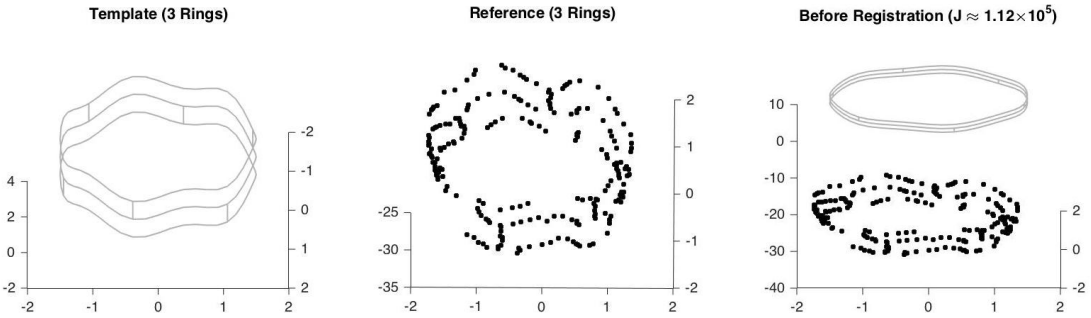

Figure 4.1: Before registration.

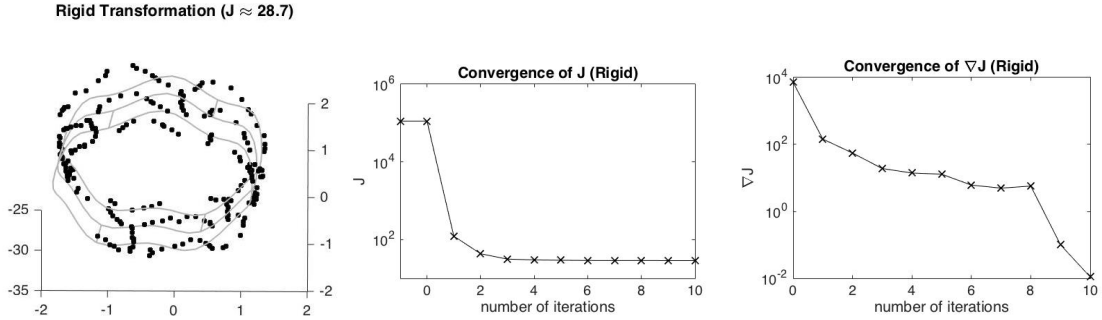

Figure 4.2: After rigid transformation.

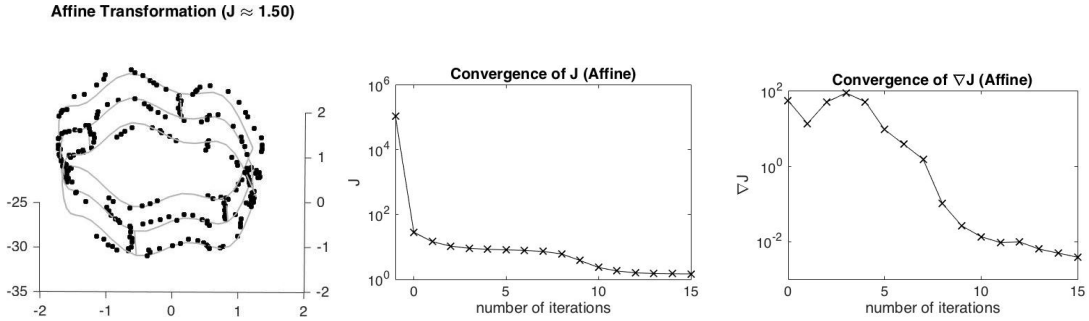

Figure 4.3: After affine transformation.

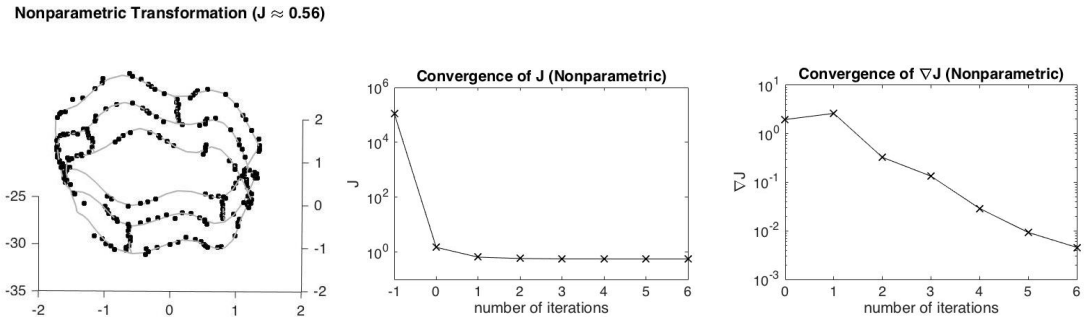

Figure 4.4: After nonparametric transformation.

misfit of $J \approx 1.50$ (Figure 4.3).

The final step is to use nonparametric transformation to further fit our template to the data. For stopping criterion here, we set the tolerance of the change in J to 10^{-4}. It takes 6 iterations, and the result is visually a better fit than that after the affine transformation, with a misfit of $J \approx 0.56$ (Figure 4.4). In this example, along with all other experiments, we use $\alpha_L = 1.2$ and $\alpha_D = 2$ as regularization parameters respectively for our length and direction regularizers, S_L and S_D. Our choice of these values is purely empirical, based on experimenting and comparing results using values of J and visual reasonability.

We observe, in each of the above three transformation steps, that the value of J decreases monotonically, and the gradient ∇J gets close to 0.

In this example, there are two steps we performed along the process: selection of the data, and rotation of the template. We left them out of the descriptions above to avoid confusion; let us now take a look at them separately.

4.2 Supplementary Procedures

In this subsection, we discuss two steps we follow as we perform our numerical experiments. These steps help improve the performance of the registration approach in our application, but they can be different in other applications.

4.2.1 Data Selection and Denoising

In this thesis, we run our experiements in a section-by-section manner: we perform registration on point clouds of selected stent sections, and combine the results to get the structure of an entire stent.

The inputs in our case are slice numbers, each of which corresponds to a point cloud of stent struts detected from an OCT slice. Therefore, selecting a section of a point

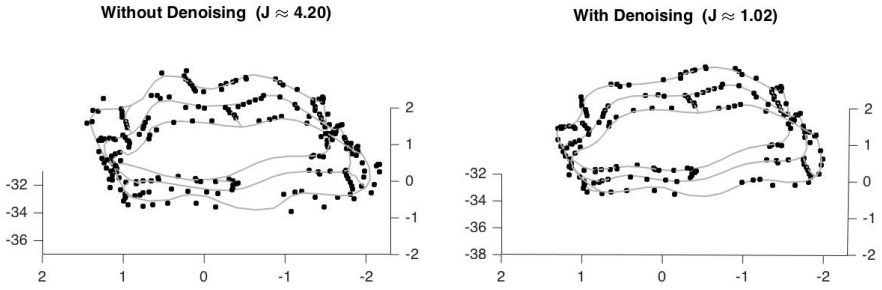

Figure 4.5: Results without and with denoising after data selection.

cloud is accomplished by simply changing the input slice numbers. An additional step we perform, after data selection, is to denoise by removing undesired data points.

Example 4.1: Denoising when Selecting Data

When we need a point cloud that corresponds to three rings, we want to remove some points from above or below these three rings (Figure 4.5).

At this stage, we perform this step of denoising manually, by interactively selecting the undesired points in the data brushing mode in MATLAB, and removing the selected data.

4.2.2 Correction of Rotation

Although rotation of the template is performed during rigid transformation, we observe undesirable results in some experiments. This is because of the non-convex nature of our problem.

To resolve this issue in our application, we replace the rigid transformation step by first translating the template using the center of mass of $\mathbb{P}$, then rotate the translated template around its first principal component, as we have done to obtain Figure 4.6. We then choose the angle corresponding to the minimal J, and use the rotated template

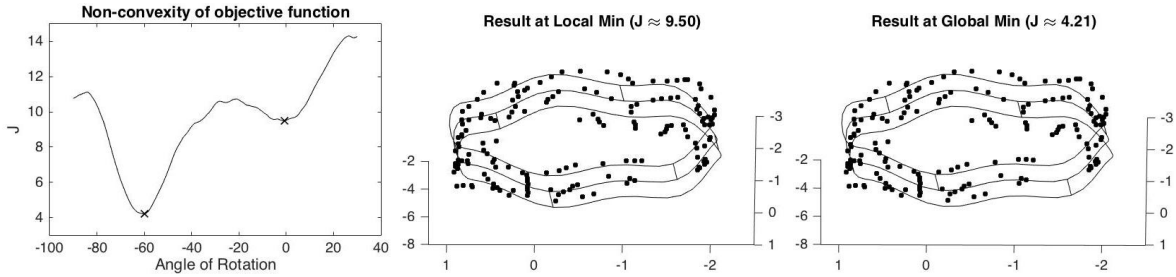

Figure 4.6: The objective function for rigid transformation is not convex, and hence solution depends on initial guess.

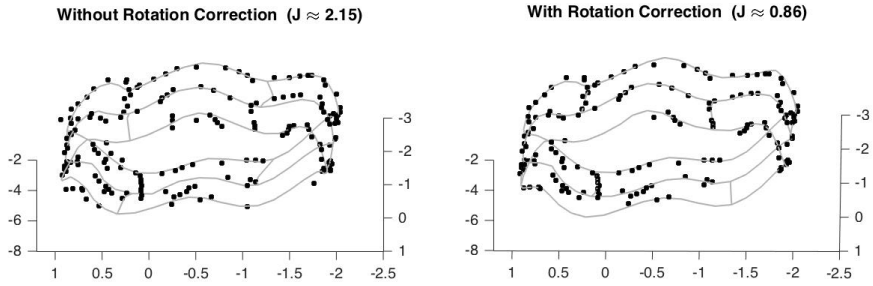

Figure 4.7: Results without and with correction of rotation.

as an initial guess for affine transformation. Performing this step, followed by affine and nonparametric transformations, gives us a reasonable result (Figure 4.7).

Example 4.2: Non-convexity of objective function

Let us choose a point cloud $\mathbb{P}$ corresponding to three rings, and translate our template by the center of mass of $\mathbb{P}$. Then rotate the translated template around its first principal component, the direction with the most variability. The resulting misfits are shown in Figure 4.6, where we observe at least two local minima. In this case, different initial guesses of the rotation angle can bring us different results.

In the above example, we use a step length of 30 degrees to find the minimal misfit, but this step length of rotation is completely dependent on the user.

Note that although we replace the step of rigid transformation in our application,

we do not remove it from the line-to-point registration scheme, because it can play an important role in more general applications, where we do not have a non-convex objective function.

4.3 Semi-Automatic 3D Registration of a Coronary Stent

Following the steps with improvements, we perform registration by sections of three to five rings, and combine the results to obtain an entire stent structure.

Figure 4.8 shows an example of a resulting stent. Given a set of observed data and knowledge of a stent structure, we can perform geometric reconstruction of different types of coronary stents (other than bioresorbable stents), using the same procedure.

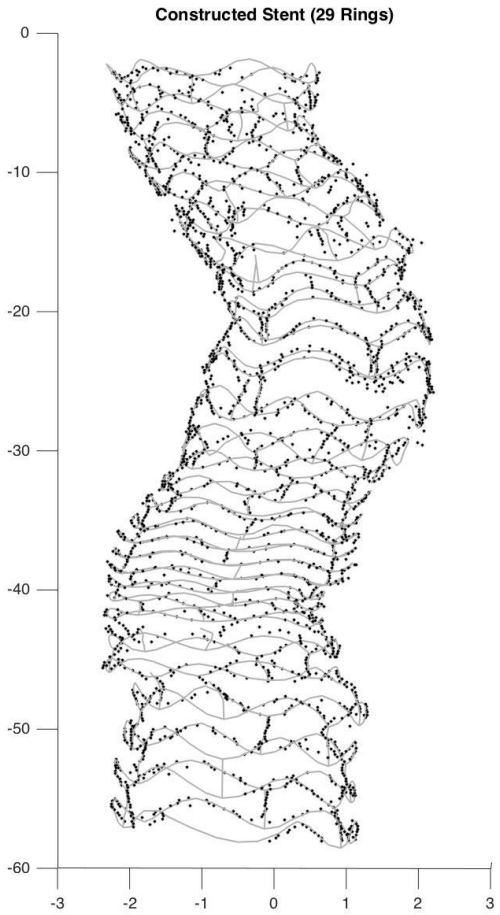

Figure 4.8: Combination of sections provide us with the structure of an entire stent.

5 Conclusions

In this section, we summarize what we have accomplished at this stage, and discuss what the next steps can be. We conclude from the viewpoints of both the general method, and our application.

5.1 Current Work

We have introduced a line-to-point registration approach, which is useful for imposing a known structure on a given set of data in 2D or 3D. This method uses the projections of points onto line segments, and aims at fitting a skeletal object with three steps of transformations. Using the length and direction regularizers, we can adjust the regularization parameters to preserve the overall structure. In general, this approach can be used in registration of two geometric objects, one of which has a skeleton structure, and the other can be represented by a point cloud.

In the application of this thesis, we use the proposed approach to reconstruct implanted coronary stents in a section-by-section manner. If categorization of data is needed, we can simply use the projection algorithm to find the closest ring or beam corresponding to each data point.

5.2 Future Directions

Method-wise, one can introduce a weight function w to the template, such that $G(\mathbb{X}, \mathbb{E}, w)$ denotes a template with weights on its nodes or edges. One can also consider adding regularizers. For example, overlapping of edges, which can result from nonparametric transformation, is unrealistic in many applications. To penalize this situation, we can impose regularization with respect to the surface areas defined by the skeletal template. However, the task of detecting surface areas, as well as the

regularization on them, is nontrivial.

In our application, we aim at a one-step registration of an entire stent. An idea is to obtain a better initial guess as the template using the centers of each OCT slice, and construct a center line to perform an initial deformation of our template. We can also utilize knowledge of the conformation of a patient's coronary arteries, to constrain the deformation of the stent skeleton.

References

[1] K. S. Arun, T. S. Huang, and S. D. Blostein. Least-squares fitting of two 3-d point sets. *IEEE Trans. Pattern Anal. Mach. Intell.*, 9(5):698–700, May 1987.

[2] M. d. Berg, O. Cheong, M. v. Kreveld, and M. Overmars. *Computational Geometry: Algorithms and Applications*. Springer-Verlag TELOS, Santa Clara, CA, USA, 3rd ed. edition, 2008.

[3] P. J. Besl and N. D. McKay. A method for registration of 3-d shapes. *IEEE Trans. Pattern Anal. Mach. Intell.*, 14(2):239–256, Feb. 1992.

[4] B. Bollobas. *Modern Graph Theory*. Springer, 1998.

[5] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, New York, NY, USA, 2004.

[6] U. Castellani and A. Bartoli. *3D Shape Registration*, pages 221–264. Springer London, 2012.

[7] Y. Chen and G. Medioni. Object modelling by registration of multiple range images. *Image Vision Comput.*, 10(3):145–155, Apr. 1992.

[8] S. Du, N. Zheng, L. Xiong, S. Ying, and J. Xue. Scaling iterative closest point algorithm for registration of m-d point sets. *J. Vis. Comun. Image Represent.*, 21(5-6):442–452, July 2010.

[9] H. W. Engl, M. Hanke, and A. Neubauer. *Regularization of inverse problems*, volume 375 of *Mathematics and its Applications*. Kluwer Academic Publishers Group, Dordrecht, 1996.

[10] E. Haber. *Computational Methods in Geophysical Electromagnetics*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2015.

[11] T. Han. Automatic strut detection of bioresorbable stent from optical coherence tomography images. Undergraduate honors thesis, Emory University, 2015.

[12] J. Modersitzki. *FAIR: flexible algorithms for image registration*, volume 6 of *Fundamentals of Algorithms*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2009.

[13] J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer, New York, 2nd edition, 2006.

[14] M. Piccinelli, L. Mirabella, T. Passerini, E. Haber, and A. Veneziani. 4d image-based cfd simulation of a compliant blood vessel. Technical report, Technical Report TR-2010-027, Emory University, 2010.

[15] B. Yang. *Numerical Modeling of Blood Flow Problems in Coronary Arteries: Patient-specific Processing, from Stented Geometries to Fluid Dynamics*. PhD thesis, Emory University, 2015.

[16] B. Yang, B. Gogas, G. Esposito, O. Hung, E. R. Arzrumly, M. Piccinelli, S. King, D. Giddens, A. Veneziani, and H. Samady. Novel in-human four dimensional wall shear stress calculation of a coronary bioresorbable scaffold using optical coherence tomography images and blood flow simulations. *Journal of the American College of Cardiology*, 65(10_S), 2015.

Symbols and Abbreviations

Symbols

$\alpha_i > 0$, *regularization parameter*

ϕ, *a transformation*

$\phi_i = \phi(x_i)$, *a point* $x_i \in \mathbb{X}$ *transformed by* ϕ

D, *distance measure*

$d = 2$ or 3, *dimension*

$\mathbb{E}$, *a set of edges in the template*

$G(\mathbb{X}, \mathbb{E})$, *a template with nodes* $\mathbb{X}$ *and edges* $\mathbb{E}$

J, *objective function*

$\mathbf{J}$, *Jacobian*

N_e, *size of* $\mathbb{E}$

N_p, *size of* $\mathbb{P}$

N_x, *size of* $\mathbb{X}$

$\mathbb{P} = \{p_1, p_2, \ldots, p_{N_p}\}$, *a point cloud of data in the reference*

$\mathbf{P}$, $\mathbb{P}$ *represented as a vector of length* $d \cdot N_p$

$\widehat{\mathbb{P}}\big(G(\mathbb{X}, \mathbb{E})\big) = \{\widehat{p}_1\big(G(\mathbb{X}, \mathbb{E})\big), \widehat{p}_2\big(G(\mathbb{X}, \mathbb{E})\big), \ldots, \widehat{p}_{N_p}\big(G(\mathbb{X}, \mathbb{E})\big)\}$, *a point cloud of projections of* $\mathbb{P}$ *onto* $G(\mathbb{X}, \mathbb{E})$

$\widehat{\mathbf{P}}$, $\widehat{\mathbb{P}}$ *represented as a vector of length* $d \cdot N_p$

S_i, *regularizer*

$\mathbf{w}$, *parameter of a parametric transformation*

$\mathbb{X} = \{x_1, x_2, \ldots, x_{N_x}\}$, *a point cloud of nodes in the template*

$\mathbf{X}$, $\mathbb{X}$ *represented as a vector of length* $d \cdot N_x$

$x_{i,1}, x_{i,2} \in \mathbb{X}$, *left and right endpoint of the* i*th edge in template* $G(\mathbb{X}, \mathbb{E})$

Abbreviations

FAIR, *Flexible Algorithms for Image Registration*

ICP, *Iterative Closest Point*

OCT, *Optical Coherence Tomography*

PCI, *Percutaneous Coronary Intervention*