NUS
National University
of Singapore

School of Computing

Doctor of Philosophy
Thesis Proposal

Predictive Surgical Simulation for Cardiac Surgery

submitted by

Li Hao
(HT050623N)

under guidance of

A/Prof. Leow Wee Kheng

January 14, 2008

Abstract

Many cardiac surgeries involve very complex operations on the heart, the great arteries and other cardiac tissues. At present, cardiac surgeons rely only on echocardiography, cardiac catherization and CT images to understand the specific anatomical structures of a patient. Without appropriate surgical planning and visualization tool, they have to resort to manual drawing to visualize the surgical procedures and the expected results. This approach is not precise and is impossible to provide detailed information about the possible outcome of the surgical procedures. To improve the precision and effectiveness of cardiac surgical planning, novel computer simulation systems that perform planning and simulation of cardiac surgeries are needed.

Among the existing surgical simulation systems, reactive simulation systems attempt to simulate real-time displacement and deformation of body tissues in response to user inputs that emulate surgical operations such as incision, resection, opening, suturing etc. They are useful for medical training and pre-operative planning of simple surgical operations. However, they are not suitable for predicting the results of complex surgical procedures. On the other hand, predictive systems produce the results of surgical procedures given a small amount of user inputs, In this way, the surgeon can explore various surgical options to determine the best ones. At present, few predictive simulation systems have been developed for open surgeries, and none for complex cardiac surgeries.

In this thesis proposal, a predictive simulation system is proposed for complex cardiac surgeries such as Arterial Switch Operation and Norwood procedure. Surgical requirements of the system are analyzed from the surgeon's perspective. These requirements are translated into system requirements that define the surgical simulation system. Candidate algorithms for achieving these requirements are discussed.

With careful analysis of the system requirements and formulation of the computational problems involved in the simulation of cardiac surgeries, a list of research tasks are identified for completing the proposed research. Some of these tasks have already been accomplished and are presented as preliminary work. The preliminary work shows that the approach adopted in this proposal is feasible and promising for developing a predictive simulation system for complex cardiac surgeries.

Contents

Chapter 1

Introduction

1.1 Motivation

Many cardiac surgeries involve very complex operations on the heart, the great arteries and other cardiac tissues. At present, the cardiac surgeons rely only on echocardiography, cardiac catherization and CT images to understand the specific anatomical structures of a patient. Without appropriate surgical planning and visualization tool, they have to resort to manual drawing to visualize the surgical procedures and the expected results. This approach is not precise and is impossible to provide detailed information about the possible outcome of the surgical procedures.

To improve the precision and effectiveness of cardiac surgical planning, novel computer simulation systems that perform planning and simulation of cardiac surgeries are needed. These systems should be able to help the surgeons to easily explore various surgical options, predict the surgical results of different options, accurately evaluate the surgical results and determine the best surgical options.

Many surgical simulation systems have been developed over the last decade. Among them, real-time simulation systems attempt to simulate real-time displacement and deformation of body tissues in response to user inputs that emulate surgical operations such as incision, resection, opening, suturing, etc [1, 3, 17, 18, 22, 24, 26, 30, 52, 55, 56, 67]. These systems often provide haptic feedback that allows the user to feel the forces while interacting with the virtual objects. They are called *reactive* systems in this thesis proposal since they react in real-time to user inputs and produce real-time simulation results.

Reactive systems are useful for medical training and pre-operative planning of simple surgical operations. However, they are not suitable for predicting the results of complex surgical procedures given a small number of user inputs. To use a reactive system to predict surgical results, the surgeon would need to go through all the surgical operations, which can be tedious and time-consuming.

Another category of surgical simulation systems is the *predictive* systems. Given a small number of necessary user inputs, predictive systems produce surgical results based on physical properties of the body tissues [4, 8, 16, 33, 35, 50, 53, 69]. Unlike real-time reactive simulation, the surgeon do not have to go through the intricate and laborious surgical operations to see the expected surgical results. In this way, the surgeon can try various kinds of surgical options and see the pros and cons of different surgical decisions quickly.

Existing simulation systems for cardiac surgery [37, 38, 55, 56] are reactive systems that simulate only simple operations such as cutting and suturing of cardiac tissues. It is very tedious and time-consuming to use such systems to perform pre-operative planning of complex surgeries such as Arterial Switch Operation and Norwood procedure. Moreover, these systems do not model the physical properties of cardiac tissues realistically. In particular, the cardiac tissues do not deform realistically under surgical operations. As a result, it is impossible to use such systems to predict the results of complex cardiac surgeries.

1.2 Research Goal

The overall goal of this thesis project is to develop a predictive simulation system for complex cardiac surgeries. To achieve this goal, it is necessary to understand the requirements of the system by studying a complex cardiac surgical procedure. For this purpose, the Arterial Switch Operation (ASO) is selected as the application example. ASO is the surgery of choice for a congenital heart disease called Transposition of the Great Arteries (TGA). Details of TGA and ASO will be discussed in Chapter 2.

Surgical simulation is performed on 3D computer models of the heart. These 3D models should be reconstructed from the patient's medical images. Reconstruction of 3D model from medical images is a challenging research topic by itself. A student in our project team is developing a sub-system for the segmentation of the heart [47]. Therefore, in this thesis proposal, the 3D models can be assumed to be given.

Based on the reconstructed 3D computer models of the heart, the system should meet the following requirements for predictive surgical simulation and planning:

1. Given a small amount of user inputs, efficiently predict the expected surgical results. The predicted surgical results should be realistic.

2. Visualize the predicted surgical results for the users, i.e., the surgeons, to explore the feasibility of their surgical options to assist them in pre-operative surgical planning.

These are the general requirements of the system. Detailed surgical and system requirements are discussed in Chapter 4.

The contributions of this research work comprise:

1. Analysis of the requirements for the predictive simulation of complex cardiac surgery.

2. Development of algorithms for predictive simulation of cardiac surgical operations.

3. Development of a predictive simulation system for complex cardiac surgeries, using ASO as a specific application example.

1.3 Organization of Thesis Proposal

To understand the difficulties and detailed requirements of the surgical simulation system, it is necessary to first discuss the anatomy and physiology of the human heart, TGA and ASO (Chapter 2). To develop the predictive simulation system, existing medical simulation systems and algorithms must be reviewed (Chapter 3). These existing systems include reactive systems (Section 3.1.1) mainly for medical training and predictive systems (Section 3.1.2) for pre-operative planning. From the computational point of view, the simulation of soft tissue deformation is the most complex part of a surgical simulation system. So existing deformable models soft tissue deformation are reviewed (Section 3.2). The proposed thesis topic and detailed requirements for the proposed system are described in Chapter 4, followed by the preliminary work (Chapter 5) illustrating the main ideas of predictive simulation for complex cardiac surgeries. Chapter 6 concludes this thesis proposal.

Chapter 2

Background

2.1 Anatomy and Physiology of Normal Heart

A normal heart consists of four chambers (Figure 2.1): left atrium, left ventricle, right atrium and right ventricle. It is a muscular organ responsible for pumping blood to the body through the blood vessels. The left side collects oxygenated blood from the lungs into the left atrium, and pumps it via the left ventricle to the body. The right side collects oxygen-depleted blood into the right atrium and pumps it to the lungs via the right ventricle.

Connecting to the heart, there are four major blood vessels: aorta, vena cava, pulmonary artery and pulmonary vein. Aorta is the main artery of the human body. It is connected to the left ventricle, and supplies oxygenated blood from the heart to the whole body. It also supplies oxygenated blood to the heart muscles through coronary arteries that are attached to it. Vena cava is attached to the right atrium. It returns oxygen-depleted blood from the whole body back to the heart. Pulmonary artery and pulmonary vein are bridges between the heart and the lungs. Pulmonary artery takes oxygen-depleted blood from the heart to the lungs, and pulmonary vein carries oxygenated blood from the lungs back to the heart.

In between the chambers and the arteries, there are four valves: mitral valve, tricuspid valve, aortic valve and pulmonary valve (Figures 2.1, 2.2). Mitral valve and tricuspid valve are also called atrioventricular valves since they are located between atriums and ventricles. Atrioventricular valves are attached to and controlled by the heart muscles. They permit one-way blood flow from the atriums to the ventricles. Aortic valve and pulmonary valve are called arterioventricular valves because they are located between arteries and ventricles. Arterioventricular valves are not directly controlled by the heart muscles. They behave in a passive manner in response to blood flow, and allow blood to pass from the ventricles to the arteries in one direction. Arterioventricular valves are also called semi-lunar valves [62] because their leaflets are shapes like a half moon.

2.2 Transposition of the Great Arteries

Transposition of the Great Arteries (TGA) [36] is a congenital heart disease in which the aorta and pulmonary artery are connected to the wrong ventricles. Unlike the normal heart, in a TGA patient, the aorta arises from the right ventricle and the pulmonary artery from the left ventricle (Figure 2.3). The coronary arteries which are in charge of nourishing the heart muscles are still attached to the aorta.

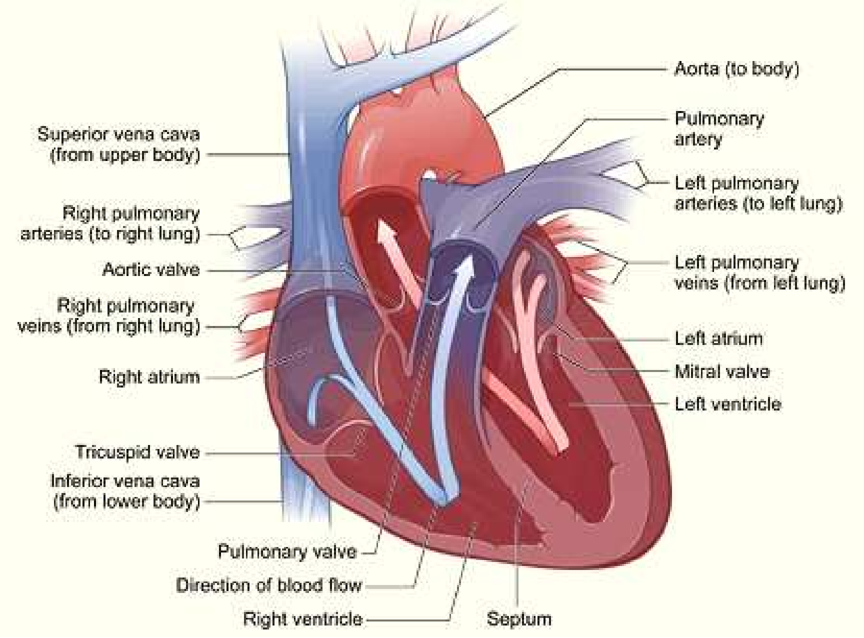

Aorta (to body)
Pulmonary artery
Superior vena cava (from upper body)
Left pulmonary arteries (to left lung)
Right pulmonary arteries (to right lung)
Aortic valve
Left pulmonary veins (from left lung)
Right pulmonary veins (from right lung)
Left atrium
Mitral valve
Right atrium
Left ventricle
Tricuspid valve
Inferior vena cava (from lower body)
Pulmonary valve
Direction of blood flow
Right ventricle
Septum

Figure 2.1: Anatomy of normal heart. Arrows indicate the directions of blood flow (from http://www.nhlbi.nih.gov/health/dci/Diseases/hhw/hhw_anatomy.html).

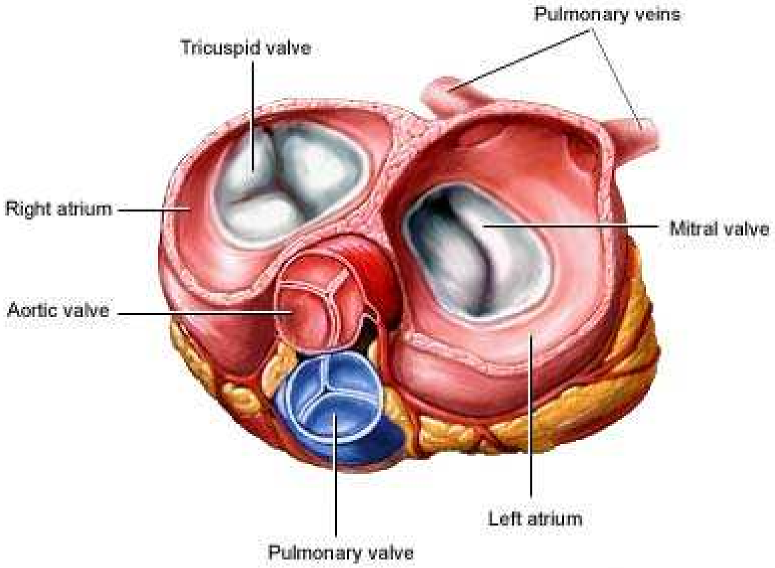


Figure 2.2: The cardiac valves (from http://www.besthealth.com/besthealth/body guide/reftext/html/cardio_sys_fin.html).

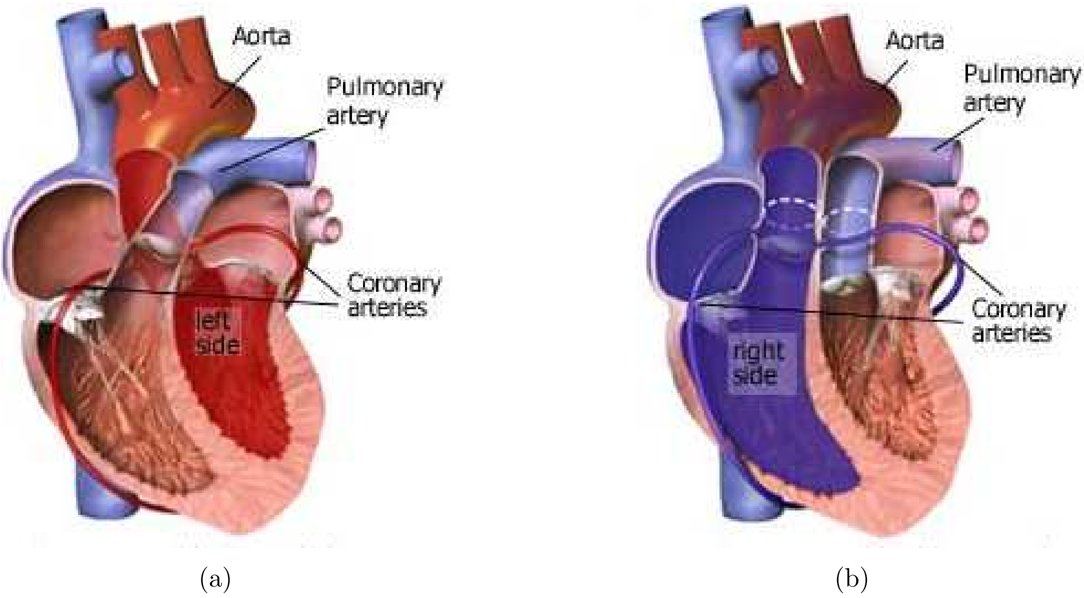


Figure 2.3: Transposition of the Great Arteries. (a) Normal heart anatomy. (b) TGA heart anatomy. The aorta and the pulmonary artery arise from the wrong ventricles (from http://heart.health.ivillage.com/signssymptoms/bluebaby3.cfm).

TGA results in a problem with blood circulation. The oxygenated blood coming from the lungs passes through the left atrium and the left ventricle, and then goes back to the lungs via the pulmonary artery. In this case, the oxygenated blood does not reach the whole body other than the lungs. On the other hand, the oxygen-depleted blood returning from the whole body goes through the right atrium and the right ventricle, and back to the whole body through the aorta without refreshing. The blood is always low in oxygen.

TGA infants are typically born with other congenital heart disease such as Ventricular Septal Defect (VSD) [20], which result in a hole or opening between the left and right ventricles. These defects allow some oxygenated blood to leak from the left ventricle to the right ventricle and then to the whole body. Without such a leakage, the infants will certainly die immediately. Even with such a leakage, the body tissues are still lack of oxygen. This results in a bluish tinge in the skin, lips and other parts of the baby, i.e., the "blue baby" symptom. The incidence of TGA is 5% of all congenital heart diseases [36]. The mortality rate of TGA in untreated patients is approximately 30% in the first week, 50% in the first month, and 90% by the end of the first year [7]. Therefore, TGA should be treated as soon as possible upon diagnosis.

2.3 Arterial Switch Operations

Arterial Switch Operation (ASO) [27, 61] is the current surgery of choice for correcting TGA. It elegantly switches the two great arteries and relocates the coronary arteries to the correct positions, restoring proper blood circulation in the patient. It is a very difficult and complex procedure that includes many delicate surgical operations. For different patients with varying anatomical structures, different variations of ASO need to be performed. It will

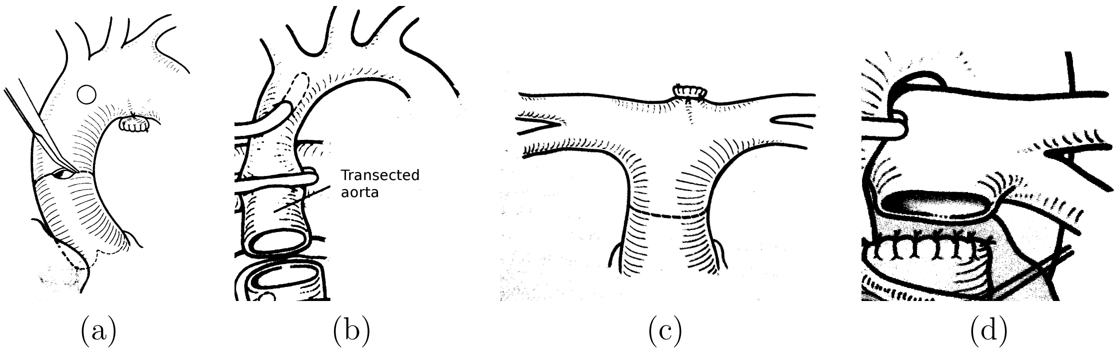


Figure 2.4: Transection of aorta and pulmonary artery [20]. (a) & (b) Transection of aorta. (c) & (d) Transection of pulmonary artery.

be too overwhelming for the readers if all the details and variations of ASO are described. So, this section just outlines the main stages of ASO. For details, please refer to [20, 27, 61].

In general, ASO contains the following four main stages [27, 61]:

1. Aorta and pulmonary artery transection.
2. Coronary artery transfer.
3. Reconstruction of neo-aorta.
4. Reconstruction of neo-pulmonary artery.

In Stage 1 (Figure 2.4), the pulmonary artery is disconnected from the pulmonary root, while the aorta is disconnected from the aortic root. This is an important stage that defines all landmarks of the arterial switch. Usually, the aorta is transected at a higher position compared with the pulmonary artery so as to reduce compression on the switched arteries.

After Stage 1, the neo-aortic root and neo-pulmonary root are to be formed (Stage 2). At this stage, the coronary artery buttons are first detached from the aortic root (harvesting of coronary buttons, see Figure 2.5(a)). Next, two pockets are cut out of the pulmonary root. Then, the coronary buttons are relocated to the pulmonary root to form the neo-aortic root (Figure 2.5(b)). After that, the aortic root is patched up to form the neo-pulmonary root (Figure 2.5(c)). Since the coronary patterns vary a lot among different patients, there are many surgical options at this stage. The main concerns are to avoid kinking of the coronary arteries and to reduce the need for suturing.

In Stage 3, the aorta is attached to the neo-aortic root in order to receive oxygenated blood coming from the left ventricle. It is usually the case that the diameter of the aorta is smaller than that of the neo-aortic root. So the aorta needs to be cut along its long axis, opened and patched with a additional tissue to enlarge its end (Figure 2.6). Sometimes, it may also happen that the diameter of aorta is larger than that of the neo-aortic root. In that case, the aorta should be tailored and sewed up to shrink its diameter.

In Stage 4, the pulmonary artery is attached to the neo-pulmonary root so that it can receive oxygen-depleted blood exiting from the right ventricle (Figure 2.7). The main consideration for performing Stages 3 and 4 is to organize the relative position of the neo-aorta and neo-pulmonary artery that the blood can flow smoothly through them. Stage 4 completes

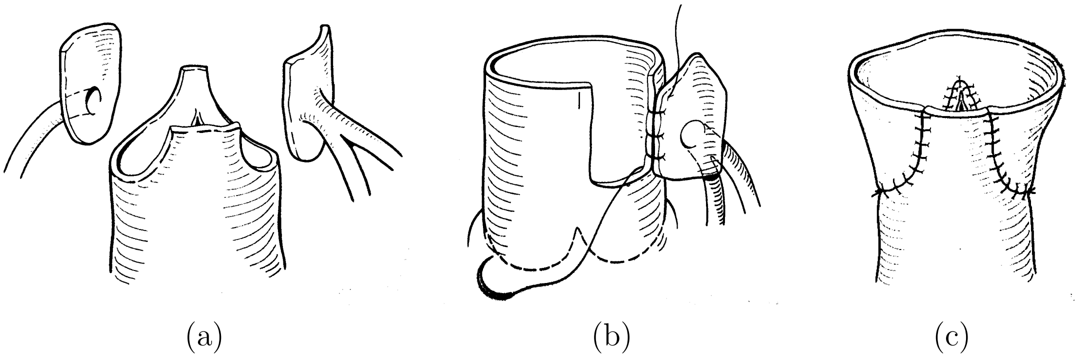


Figure 2.5: Coronary artery transfer [20]. (a) Harvesting of coronary buttons from the aortic root. (b) Relocation of the coronary button to the pulmonary root to form the neo-aortic root. (c) The aortic root is patched up to form the neo-pulmonary root

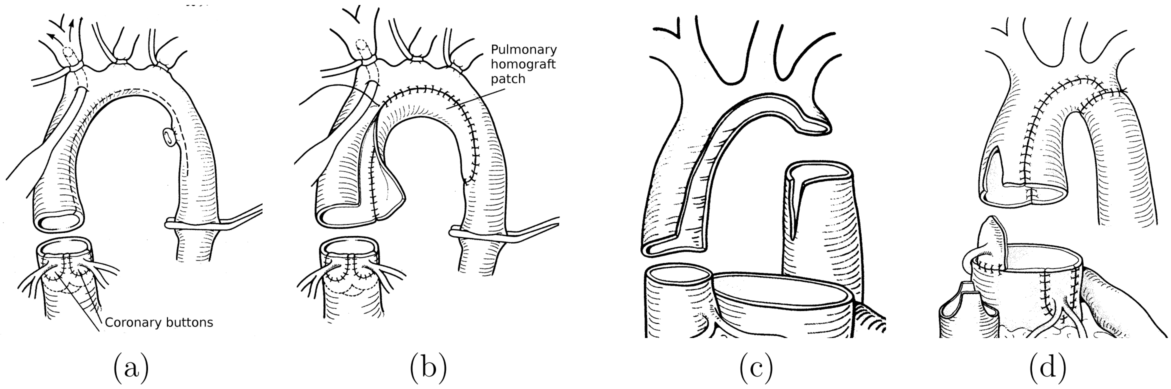


Figure 2.6: Reconstruction of aorta [61]. (a) The aorta is cut longitudinally (dashed curve). (b) The aorta is opened and patched to match the diameter of the neo-aortic root. (c) The aorta is cut longitudinally and opened up. (d) The aorta is patched to match the diameter of the neo-aortic root.

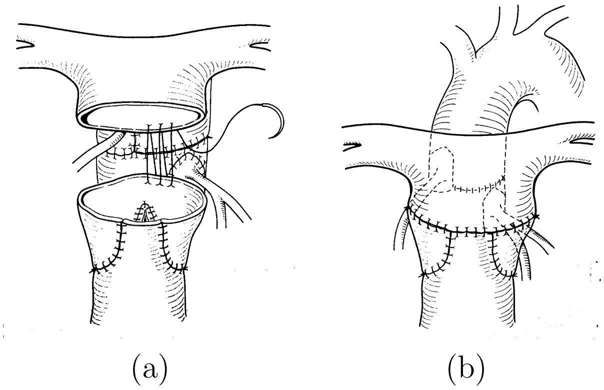


Figure 2.7: Reconstruction of pulmonary artery [20].

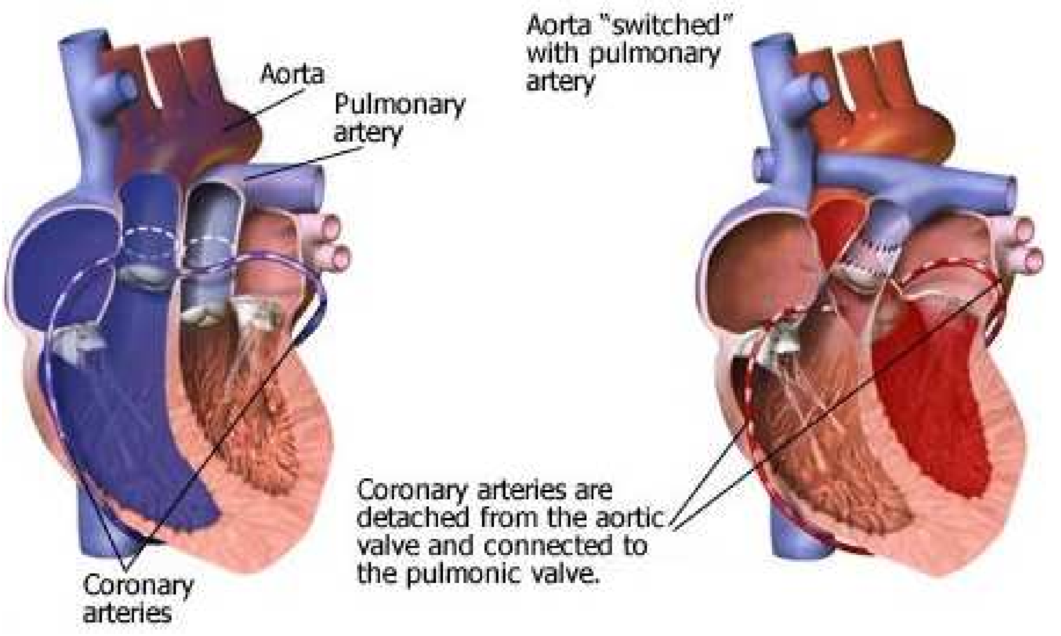

Aorta
Pulmonary artery
Coronary arteries
Aorta "switched" with pulmonary artery
Coronary arteries are detached from the aortic valve and connected to the pulmonic valve.

Figure 2.8: The anatomical strucuture of a TGA heart before (left) and after (right) ASO (from http://heart.health.ivillage.com/pediatricheartsurgery/arterialswitch.cfm).

the correction of the wrongly attached arteries. A comparison of the heart structure before and after ASO is shown in Figure 2.8.

ASO is elegant and able to offer longer life expectancy to the patient. There are less wounds made in ASO than in other types of surgeries for TGA. So the chances of rhythm disturbances caused by later scaring are reduced.

ASO is a difficult and complex surgery due to the complexity of the anatomical structure of the heart. The patients are infants, whose arteries, especially coronary are very thin (about 1–2 mm in diameter). As a result, the operations require delicate surgical techniques with very low safety margins. Coupled with lower birth rate and high social expectation, providing quality health care to the public becomes an important issue of ASO.

Chapter 3

Related Work

Over the last decades, many surgical simulation systems have been developed for various surgeries, anatomical structures and purposes. To analyze the state of the art of surgical simulation, these existing systems are reviewed (Section 3.1). Most surgical simulation systems require the simulation of soft tissue deformation, which is usually one of the most complex parts in a simulation system. Thus, existing 3D deformable models that can be used for surgical simulation are also discussed in this chapter (Section 3.2).

3.1 Surgical Simulation Systems

In this section, existing surgical simulation systems are reviewed. According to their characteristics, they are categorized into reactive systems and predictive systems.

3.1.1 Reactive Systems

Reactive systems are real-time simulation systems that attempt to simulate the reactions (i.e., displacement and deformation) of body tissues in response to user inputs that emulate surgical operations such as incision, resection, opening, suturing, etc. The objective of reactive systems is to provide the user with realistic situations and perception of surgical procedures, through user interactions and simulated progressive behaviors of the body tissues.

Reactive systems have been developed for needle-based procedures [22, 24, 29, 65]. These systems simulate body tissue deformation in response to needle insertion. Because needle insertion is a simple and straightforward surgical operation, reactive systems for needle-based procedures are well developed, and can provide real-time visual and haptic feedback during simulation at low computational cost.

Reactive systems have also been developed for the simulation of minimally invasive surgeries (MIS). MIS introduce specialized instruments into the body through small incisions to manipulate the body tissues. Visual feedback is obtained from the cameras attached to the inserted instruments. Constrained by the small entry portal, the instruments are usually designed to work in a limited range of motion.

Reactive systems for MIS [5, 17, 26, 52, 58] usually model interior anatomies and generate organ surface features using medical images. They simulate real-time soft tissue deformation in response to the limited motion of the instruments, and provide visual and haptic feedback to the user. The limited motion of the instruments makes the surgical operations relatively simple, which facilitates the development of reactive systems for MIS. However, when the anatomical structures to be dealt with involve complex soft tissues, achieving real-time and realistic soft tissue deformation is still difficult. This becomes the main challenge of reactive system for MIS.

Compared to needle-based procedures and minimally invasive surgeries, open surgeries are more complex and difficult to simulate because of the larger visual field, complex anatomical structures, and the freedom of motion of the surgical instruments [1]. As a result, reactive systems for open surgeries have focused only on simulating simple basic surgical operations such as incision and soft tissue retraction [3, 55, 56], suturing of wounds [67] and ventricular septal defect [55, 56]. In comparison, simulation of cardiac surgeries such as ASO and Norwood procedure is far more complex and difficult to be achieved in real time. Some reactive systems have simulated the behavior of body parts such as intestine in response to touch and displacement by a surgical instrument [18].

Modeling realistic real-time behaviors of soft tissue is a computational expensive task. One method for reducing processing time is to perform the computations in graphics processing unit (GPU) [44, 48, 54]. This approach has been applied to the simulation of simple cardiac surgery [38, 39].

Reactive systems are useful for medical training and pre-operative planning of simple basic surgical operations such as cutting and suturing. They are not suitable for pre-operative planning of complex surgical procedures such as ASO and Norwood procedure. To use a reactive system to predict the surgical results of complex procedures, one would need to go through all the delicate surgical operations in the procedures, which is tedious and time-consuming. Moreover, it is difficult to achieve realistic and real-time deformation of complex anatomical structures.

3.1.2 Predictive Systems

Predictive systems attempt to accurately predict surgical results of complex surgical procedures based on a small number of user inputs. They allow the users to easily explore various surgical options to access the feasibility of each option and to choose the best one. Compared to reactive systems, predictive systems are designed to simulate entire surgical procedures instead of low-level basic surgical operations. In predictive systems, real-time response is not a necessary requirement. Instead, the accuracy of the predicted results becomes very important.

Predictive simulation systems have been developed for pre-operative planning of dental surgeries and surgical implants [16, 19, 59, 66]. They have also been developed for predicting post-operative alignment of bones in high tibial osteotomy [64] and surgical repositioning of fractured radius (arm bones) that is not properly fixed [50]. In these applications, the anatomical parts are all rigid objects such as bones and teeth.

Surgical planning systems have also been developed for predicting facial appearance due to displacement of bones [21, 6, 35, 72]. The deformation of facial soft tissues is achieved using Finite Element Method (Section 3.2.2). This method algorithm has also been applied on the prediction cornea deformation due to eye muscle forces and pressure [13].

For liver surgery, [4, 33] developed systems that analyze segmentation results of CT images to propose liver and tumor resection (i.e., cutting) plan based on domain knowledge about the safety margins of the operations. The system in [33] also estimates the postoperative liver volume. For breast reconstructive surgery, [8, 69] developed planners that predict the postoperative shape of the breast using predefined breast models.

To date, there is no predictive systems for simulation and pre-operative planning of complex cardiac surgeries.

3.2 3D Model Deformation

There are four main approaches to 3D object deformation: free-form deformation, mass spring model, Finite Element Method, and differential geometry method. Free-form deformation (FFD) [11, 31, 51] modifies an object's shape implicitly by deforming the 3D space that contains the object. Since it does not work directly on geometric shape, it is difficult for FFD to manage geometric properties that are defined on mesh vertices, edges, and faces. Therefore, it is inappropriate for surgical simulation. On the other hand, the other three approaches modify an object's shape explicitly such that geometric properties can be incorporated easily. They are suitable for surgical simulation.

3.2.1 Mass Spring Model

Mass spring model (MSM) represents the 3D mesh as a network of mass points connected by massless spring-dampers [26, 28, 37, 38, 45]. It models external forces acting on the object, which are opposed by internal spring forces. Numerical solution of mass spring equation gives the position of each point, thus defining the shape of the object. MSM can be applied on both surface mesh and volumetric mesh.

MSM is useful for real-time surgical simulation due to its simplicity in implementation and efficiency in computation. With an appropriate numerical solution and a good implementation, a MSM can be fairly large and complex without compromising real-time response. Therefore, MSM is appropriate for real-time surgical simulators.

Padilla Castañeda et al. [45] used mass spring model to build a prostate model. Their model can simulate transurenthral resection of the prostate and tissue deformation due to the resection in real time. Choi [9] and Kuhnapfel et al. [26] applied MSM to support simulation of interactive cutting on deformable anatomy. Lian et al. [28] presented a simulation of progressive suturing where MSM was used to provide both visual and haptic rendering. Mosegaard [37] applied MSM for real-time cardiac surgical simulation. Their model consists of both outer surface and inner suface of the heart. Additional springs were inserted in between the two surfaces to simulate the forces between surfaces. Sørensen et al. later extended this model for simulating more complex surgical operations using GPU [38].

Although MSM is computationally efficient and easy to implement, it has several drawbacks. Simulation using MSM is not necessarily realistic. The model's behavior depends highly on its topology and resolution. There is no standard way to identify a proper mesh that can react realistically towards different kinds of surgical operations. Bianchi et al. [2] presented a genetic algorithm for identifying the topology of MSM by comparing its behavior with a known reference model. But in practice, acquiring a realistic reference model itself is a problem. Another drawback of MSM is that soft tissues can have non-linear and inhomogeneous elasticity. Specifying realistic elasticity value at each mesh vertex is a difficult task.

Numerical integration methods for solving MSM equations are not always stable. The stability of the solutioin depends on the time step and the integration scheme [41]. In general, the condition for numerical stability is that the time step should be smaller than a stability threshold, which depends on the stiffness of the simulated object and the numerical integration scheme. An analysis of the stability conditions for different numerical methods is given in [23].

3.2.2 Finite Element Method

Finite Element Method (FEM) formulates soft tissue deformation in a continuous domain. In FEM, the object is divided into volumetric elements, and the deformation of the object is represented by an energy function derived from the external forces applied on these elements, which are opposed by the object's internal energy defined by stress and strain. The object reaches its equilibrium shape when its potential energy reaches a minimum. FEM is designed for volumetric meshes. To use FEM for surface meshes, volumetric representations should be generated and coupled with the surface meshes [40].

By formulating the deformation problem in a continuous domain and solving it in a discrete way, FEM gives a more accurate and realistic modelling of soft tissue deformation than MSM. There are many applications that use FEM to simulate deformation in surgical simulation. Crouch et al. [13] used FEM to simulate cornea deformation for cataract surgery. Richens et al. [49] used FEM to investigate the behavior of aorta during blunt traumatic impacts. It included a thorax and heart structure model that simulated the response of the heart following a thoracic impact. In addition, a detailed model of the heart and aorta was used to simulate the stresses acting on the aortic isthmus during the thoracic impact. FEM was also used to analyze left ventricular wall stress in patients with severe aortic insufficiency [70].

Although FEM can be used to provide a more realistic simulation than MSM, it is computationally expensive. Moreover, when the shape or topology of the object changes, the model may need to be re-computed. By reducing the number of elements or simplifying the representation of the internal energy, the computational cost can be reduced. [12, 71] tried to reduce the number of elements using an adaptive coarse-to-fine model. Nienhuys et al. [42, 43] presented an iterative algorithm for an interactive deformation simulation caused by cutting. These methods typically trade model accuracy for processing speed.

3.2.3 Differential Geometry Method

Differential geometry (DG) method, which is known as partial differential equation (PDE) method in [32], directly solves for the positions of the mesh points after deformation, given the initial configuration of the object and predefined constraints on the geometric properties of the mesh surface. Depending on the constraints, the system equations may be solved using a linear or non-linear solver.

DG is widely used in computer graphics community for mesh editing purposes [15, 32, 34, 57, 60, 73]. Unlike MSM and FEM, DG does not explicitly model external physical forces acting on the object. Therefore, it has not been used in surgical simulation. However, for predictive surgical simulation, the surgeons are interested in the expected surgical results instead of continuous shape change due to human interaction. Therefore, DG turns out to be very useful for predictive simulation, and is adopted in our research for the simulation of complex cardiac surgery (Chapter 5).

DG is easy to implement and is relatively fast compared to FEM. It can achieve real-time deformation for 2D meshes [68] and near real-time deformation for 3D meshes [32].

Various geometric constraints can be incorporated into the deformation scheme. However, it is challenging to translate elastic properties of materials to geometric properties of the mesh.

The standard DG method applies Laplacian operators to estimate the mean-curvature normals of the surface. As the Laplacian operators are ill-defined on mesh boundaries, existing Laplacian methods assume that the models are 2-manifold surfaces without boundaries.

Another DG method is to model the mesh surfaces using Poisson method [46, 73, 74], which describes the differential properties of mesh surfaces in terms of continuous partial differential equation. It can, therefore, model smoothness and continuity naturally. However, the Poisson method requires boundary conditions on all boundaries, which may not be known as a priori in our application. Moreover, it is more complex and tedious to implement than the Laplacian method.

3.3 Summary

Reactive simulation systems are useful for medical training and pre-operative planning of simple surgical operations such as cutting and suturing. Predictive simulation systems are suitable for pre-operative planning of complex surgical procedures such as ASO and Norwood procedure. Existing systems for simulating cardiac surgeries perform only basic surgical operations [37, 38, 39, 55, 56] (Table 3.1). There is no system for simulating complex cardiac surgical procedures, whether reactive or predictive.

For simulation of soft tissue deformation, three deformable models have been discussed. Among them, MSM is efficient and easy to implement, but it is not suitable for providing realistic deformation results. It is usually used for reactive real-time simulation. FEM can be used for realistic simulation of soft tissue deformation. However, it is computationally too expensive for real-time or near real-time simulation. On the other hand, DG can be used to produce more realistic simulation results than MSM, and it is computationally more efficient than FEM. Therefore, DG is a promising candidate for predictive simulation of complex cardiac surgery, and is adopted in the proposed research.

Table 3.1: Summary of existing surgical simulation systems and deformation methods used.

	Reactive	Predictive
Rigid model	——	• bone alignment [50, 64] • reconstructive dental surgeries [16, 19, 59, 66]
Mass spring model	• basic surgical ops. [3, 28, 67] • minimally invasive surgeries [26] • open surgeires: – heart [37, 38, 39, 55, 56] – intestine [18]	——
Finite Element Method	• basic surgical ops. [42, 43] • needle-based surgeries [22] • minimally invasive surgeries [52, 58]	• facial plastic surgeries [6, 21, 35, 72] • cornea surgery [13]
Differential Geometry method	——	——
Other systems	• needle-based surgeries [24, 29, 65] • minimally invasive surgeries [5, 17]	• breast reconstruction [8, 69] • liver tumor resection [4, 33]

Chapter 4

Proposed Thesis Project

The overall research goal of the thesis project is to develop a predictive surgical simulation system for general cardiac surgeries. To achieve the goal, surgical requirements of predictive simulation should be first understood (Section 4.1). These surgical requirements must be translated into system requirements (Section 4.2) to define the surgical simulation system. Candidate algorithms for implementing these requirements and possible validation methods are then discussed (Sections 4.3 and 4.4). Finally, a tentative research schedule for the thesis work is presented (Section 4.5).

4.1 Surgical Requirements

Surgical requirements refer to the requirements of the surgical simulation system from the surgeon's (i.e., user's) perspective. These requirements are gathered by discussing with a cardiac surgeon and referring to cardiac surgical handbook [20]. In general, the term "surgical operation" can be used to refer to either basic operations such as cutting and suturing, or complex surgical procedures such as Arterial Switch Operation (ASO) and Norwood procedure. For clarify, three levels of surgical operations are defined in this thesis proposal (Figure 4.1): low-level basic surgical operations, mid-level surgical tasks and high-level surgical procedures.

4.1.1 Low-level Requirements

Low-level basic surgical operations refer to simple operations that form the basis of all complex operations. They include cutting of tissue, displacement of tissue, and joining of two tissues together.

A. Cutting of Tissue

Cutting of tissue consists of three cases: incision of tissue, transection of artery, and resection of tissue. Incision of tissue is to cut a slid on the tissue to open it. After this operation, a cut is made on the tissue surface, but the tissue is not split into two separate parts. For example, in Figure 4.2(a), the cut is made along the aorta longitudinally. Transection of artery is to perform a cross-sectional cut of the artery (Figure 4.2(b)). After cutting, the artery is split into two segments. Resection of tissue is to cut out a piece of tissue from an anatomical part. The isolated tissue is usually joined to other anatomical parts using the

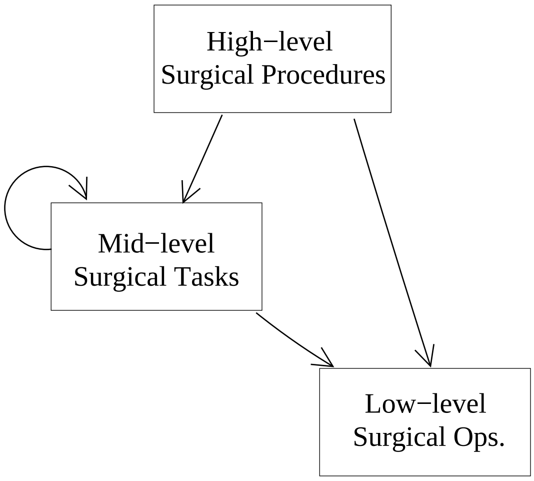


Figure 4.1: Surgical operation hierarchy. Arrow means "involves".

join operation. Figure 4.2(c) shows an example of resection of a piece of tissue from the aortic root.

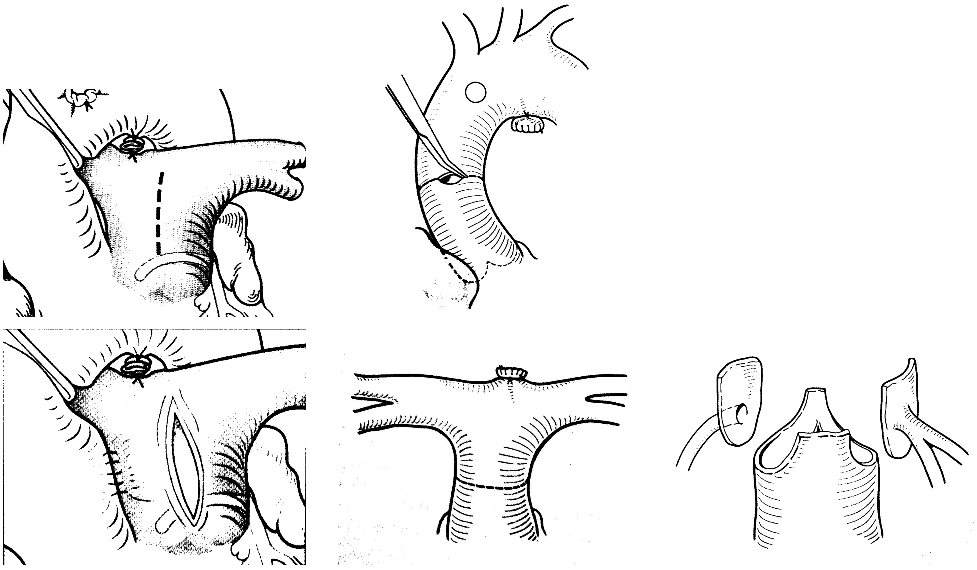
Note that in real situations, soft tissues will deform in response to cutting operation due to tension on the tissue surface (Figure 4.2(a)). On the other hand, our system does not have to be reactive, and cutting is never the last operation that will result in deformation. So, it is not necessary to handle soft tissue deformation in cutting operation. Therefore, the surgical requirement for cutting is only to create a cut on the object without deforming the object immediately. The user input for cutting is a cut path on the tissue's surface and the type of cut.

B. Displacement of Tissue

Displacement operation refers to the movement of an object or a part of it from one position to another. There are three cases of displacement. The first case is performed on the incision result. For example, in opening of aorta, the cut boundary are displaced to open the aorta (Figure 4.3(a)). The second case is the displacement of transection result. For example, in Stage 4 of ASO (Section 2.3), one end of the pulmonary artery is moved to join with the neo-pulmonary root (Figure 4.3(b)) while the other ends are fixed. The third case is the displacement of the result of resection. An example is to move the coronary buttons disconnected from the aortic root to their new positions in the pulmonary root to form the neo-aortic root.

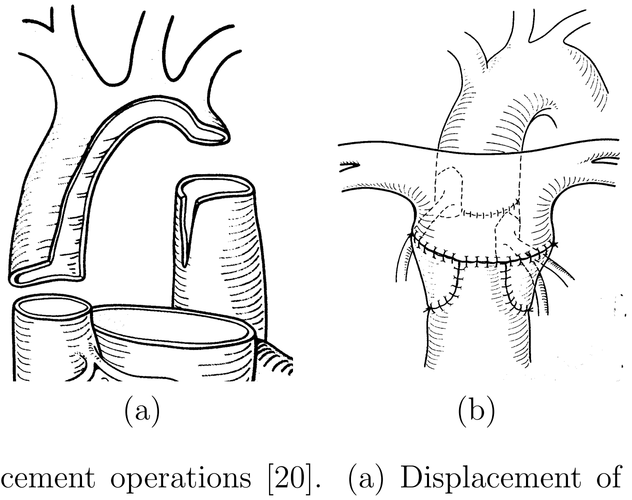
For the first two cases, only some parts of the artery are displaced while the other parts of the artery are fixed. These kinds of displacement result in soft tissue deformation. For the third case, the isolated object (e.g., coronary button) is moved so that there is no soft tissue deformation in the displacement. In this case, the tissue can be regarded as a rigid object.

The user inputs for displacement operation include the object or part of the object to be moved, and the target position and orientation of the object. Note that displacement is the main cause of soft tissue deformation, and it is one of the most common low-level operations. So it is natural to handle soft tissue deformation while simulating displacement.



(a) (b) (c)

Figure 4.2: Cutting of tissues [20]. (a) Incision of a tissue. (b) Transection of an artery. (c) Resection of a tissue.



(a) (b)

Figure 4.3: Displacement operations [20]. (a) Displacement of incision result. (b) Displacement of transection result.

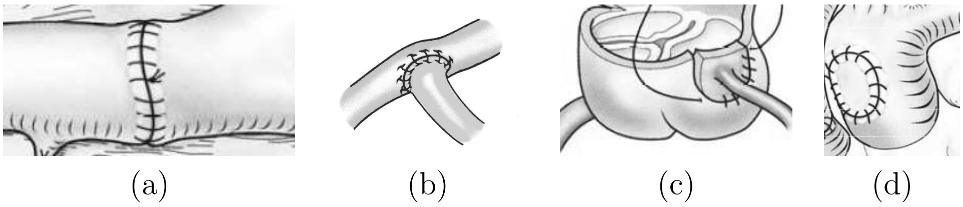


Figure 4.4: Joining operations [20]. (a) End-to-end joining. (b) End-to-side joining. (c) & (d) Side-to-side joining.

C. Joining of Tissue

Joining operations are performed to connect two tissues at their boundaries. It can be achieved through suturing or gluing. Detailed simulation of suturing involves the interaction of the tissue with needle and thread, which is very complex. For gluing, the two joining boundaries may overlap. These details will not affect the overall surgical results significantly. Therefore, they are omitted in this thesis proposal. Instead of detailed simulation of suturing and gluing, this thesis proposal considers the joining of the boundaries of two tissues as an adequate approximation.

There are three common cases of joining operation in cardiac surgeries, namely end-to-end joining, end-to-side joining, and side-to-side joining. End-to-end joining is to join two arteries at their ends (Figure 4.4(a)). It occurs in ASO when joining the aorta to the neo-aortic root, and joining the pulmonary artery to the neo-pulmonary root. End-to-side joining is to join the end of an artery to the side of an artery. Figure 4.4(b) is an example of end-to-side joining in Norwood procedure. Side-to-side join refers to joining a relatively flat patch with a tissue to fill up a hole or slot on that tissue. For example, in ASO, side-to-side joining with open joining boundaries occurs in the stage of relocating coronary buttons. The coronary buttons are joined to slots cut out of the pulmonary root (Figure 4.4(c)). Figure 4.4(d) illustrate another case of side-to-side join where an hole on one artery is completely patched up by a piece of tissue.

Note that the two joining boundaries are not necessarily of equal length. In real surgery, if the length mismatch is small, the tissue with a shorter boundary can be stretched to fit the one with a longer boundary. On the other hand, if the length mismatch is too large, the one with a shorter boundary is patched with an additional tissue to match the one with a longer boundary (Figure 4.5(d)). So, joining operation can result in the deformation of soft tissues.

Another consideration is whether the join should be smooth. In Figure 4.4(a), (c) and (d), the joins should to be smooth to avoid stenosis (obstruction of blood flow). On the other hand, in Figure 4.4(b), the join is not a smooth join.

4.1.2 Mid-level Requirements

A mid-level surgical task consists of a sequence of low-level basic surgical operations and possibly other mid-level surgical sub-tasks. In ASO for example, these surgical tasks include: transfer of coronary arteries, opening of artery, patching of artery, and reconstruction of artery.

Let us consider, for instance, transfer of coronary arteries (Figure 4.5(a–c)). The coronary

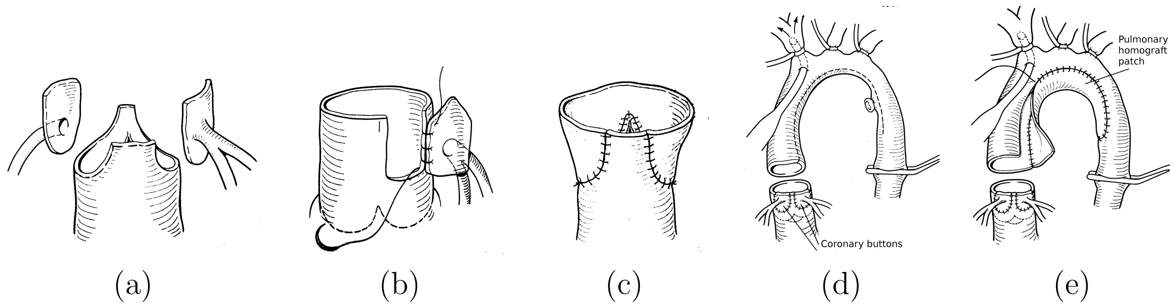


Figure 4.5: Mid-level surgical tasks [20]. (a–c) Transfer of coronary arteris. (d, e) Reconstruction of neo-aorta.

buttons are first removed from the aortic root (resection). Next, two slots are cut out of the pulmonary root (resection). Then, the coronary buttons are attached to the pulmonary root (displacement and side-to-side joining). Finally, the two slots on the aortic root are patched up (displacement and side-to-side joining). The joins in this case should be smooth.

In the case of reconstruction of neo-aorta (Figure 4.5(d, e)), the aorta is joined to the neo-aortic root (displacement and end-to-end joining). If the diameter of the aorta is too small compared to that of the neo-aortic root, the aorta has to be opened (opening of aorta) and patched with additional tissue (patching of artery) (Figure 4.5(e)). In this case, reconstruction of neo-aorta involves other mid-level sub-tasks including opening and patching of artery. In a similar way, other surgical tasks can involve low-level basic surgical operations and mid-level surgical sub-tasks.

4.1.3 High-level Requirements

High-level surgical procedure refers to the complete procedure of a cardiac surgery. Simulation and prediction of high-level surgical procedures are the ultimate goals of the predictive system. A high-level surgical procedure can involve both mid-level surgical tasks and low-level basic surgical operations. For instance, ASO consists of four main stages:

1. Transection of aorta and pulmonary artery.
2. Transfer of coronary arteries.
3. Reconstruction of neo-aorta.
4. Reconstruction of neo-pulmonary artery.

The first stage involves low-level basic surgical operations whereas the other stages involve mid-level surgical tasks as discussed in Section 4.1.2.

Another example of high-level surgical procedure is Norwood procedure. Norwood operation is the surgery of choice for Hypoplastic Left Heart Syndrome (HLHS). In a child with HLHS, all the structures on the left side of the heart (the side that receives oxygen-rich blood from the lungs and pumps it out to the body) are severely underdeveloped. The mitral and aortic valves are either very small or completely closed. The left ventricle itself is tiny

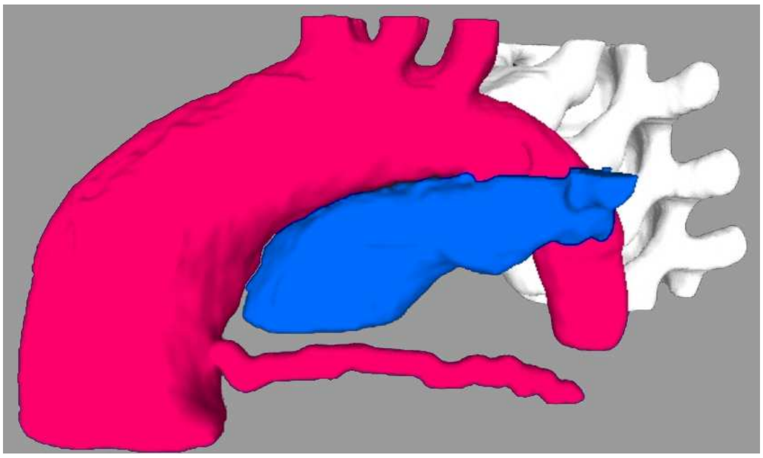


Figure 4.6: 3D mesh models of aorta, pulmonary artery and coronary artery reconstructed from patient's CT images. The spine (shown in white) is also reconstructed for orienting the 3D models (from [47].

(hypoplastic left ventricle), and the first part of the aorta is very small, often only a few millimeters in diameter (hypoplastic ascending aorta and aortic arch).

Norwood operation consists of four main stages:

1. Transection of aorta and pulmonary artery (similar to ASO).
2. Closure of pulmonary artery.
3. Reconstruction of neo-aorta (similar to ASO).
4. Placement of a tube graft to connect aorta to pulmonary artery.

Each of these stages involves a mid-level surgical task. For details of Norwood operation, refer to [10].

4.2 System Requirements

System requirements for the predictive simulation system are the requirements from the system's point of view. They are derived from the surgical requirements. The three levels of surgical requirements are translated into three levels of system requirements respectively.

The 3D mesh models of the blood vessels of interest are assumed to be already reconstructed from the patient's medical images. As the blood vessels are very thin, especially in infants, their thickness are ignored in this research. That is, the blood vessels are modeled as 2-manifold triangle meshes. A 2-manifold triangle mesh is a triangle mesh whose edges are either manifold edges or boundary edges. A manifold edge is an edge that has exactly two neighboring triangular faces. A boundary edge has one only one neighboring face. There are no isolated vertices in a 2-manifold mesh. Sample mesh models of the aorta, pulmonary artery and coronary arteries reconstructed (by our project team member) from patient's medical images is shown in Figure 4.6.

4.2.1 Low-level Requirements

Low-level surgical operations are translated into mesh operations, which include mesh cutting, mesh displacement and deformation, and mesh joining (Table 4.1).

A. Mesh Cutting

The three cases of cutting operation translate to three cases of mesh cutting. In the case of incision (cutting a slid), two cut edges coinciding with the user-specified cut path are created on the mesh model. In the case of transection (cross-sectional cutting of artery), the mesh model of the artery is split into two parts, and two cut boundaries are created, one on each part. From the system's point of view, the case of resection is similar to that of transection except that the resection result is an isolated mesh model.

The inputs for mesh cutting operation include the type of the cut and the cut path. For incision and resection of tissues, the cut path can be manually drawn on the mesh surface or specified by a sequence of points. In the latter case, the system should automatically compute the shortest smooth curve that pass through all the points. The curve is then taken as the cut path. For transection of an artery, the input can be only one landmark point indicating the position of the cross-sectional cut.

B. Mesh Displacement and Deformation

Surgical displacement operation translates to mesh displacement and deformation. Displacement of isolated mesh model (i.e., resection result) involves only translation and rotation of the mesh model. In this case, the mesh model can be regarded as a rigid object. On the other hand, displacement of non-isolated mesh model (e.g., incision and transection results) entails the deformation of the mesh model because part of the model is displaced while the other parts are fixed.

The user inputs of mesh displacement operation include the mesh model or part of the mesh model to be moved, and the target position and orientation of the mesh model. These inputs are considered as positional constraints of the mesh models during mesh deformation.

For realistic prediction of soft tissue deformation, accurate modeling of the elastic properties are very important. This is a very difficult and challenging research topic by itself. Therefore, this thesis work will just seek an approximate modeling method that is efficient for simulation. The system would provide intuitive GUI to allow the user to vary elastic properties, e.g., from very rigid to very elastic. Then, the users can easily tune the elasticity of the mesh models to match their surgical experience.

The system also needs to check whether the user-specified displacement can be achieved. For this purpose, the system should investigate (1) the geometrical shape and size of the mesh models, (2) the elastic properties of the mesh models and (3) obstructions among mesh models.

For example, if the target position is too far away from the original position, and elasticity is limited, then the displacement cannot be achieved. The free space for displacement is small because the arteries and the heart are quite tightly packed into a small space (Figure 4.7). The deformed mesh models should not penetrate each other.

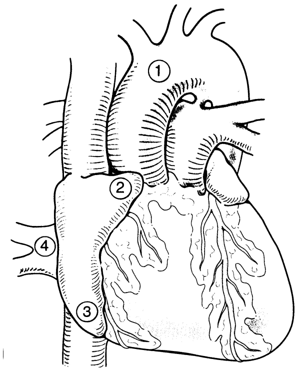


Figure 4.7: The arteries the heart are quite tightly packed into a small space [20].

C. Mesh Joining

Surgical joining operation corresponds to mesh joining. Given the joining boundaries and the type of join, mesh joining operation places the two joining boundaries together forming a seamless join.

Typically, the original positions of the two joining boundaries do not coincide with each other. So mesh deformation may be involved in mesh joining operation to join the joining boundaries. Sometimes the joins should be smooth (Figure 4.4(a), (c) and (d)). The feasibility of the joining operation should also be checked, just like in mesh displacement operation.

Note that the two mesh models that are joined may or may not be merged into a single mesh. If subsequent operations need to manipulate the joined models as a single object, then they should be merged. Otherwise, it is not necessary to merge them.

4.2.2 Mid-level Requirements

Mid-level surgical tasks are translated into their corresponding mid-level computational tasks. A mid-level computational task needs to be formulated as a computational problem, which is solved by an algorithm that performs a sequence of low-level mesh operations and possibly mid-level sub-tasks. For example, the reconstruction of aorta will be formulated as a problem that involves the opening of aorta and the shape completion of aorta, which are in turn, formulated as respective computational problems. Details of these problem formulations are presented in Chapter 5.

Depending on the nature of the problem, various constraints need to be imposed on the deformation of the mesh models. For example, after soft tissue deformation, the shape of the deformed tissue should remain similar to the original ones. The deformed mesh models should not penetrate other meshes models. Another constraint is the stretching and bending constraint which is used to control the elasticity of the object. Usually, the stretching and bending should be minimized to prevent stenosis. In some cases, the join of two mesh model should be smooth.

The constraints are summarized as follows:

Constraint A. Positional constraint. The target positions of some parts (i.e., vertices, edges and faces) of the mesh model are ether specified by the user or derived from user inputs.

Constraint B. No penetration constraint. The deformed mesh models should not penetrate other meshes.

Constraint C. Shape constraint. The shape of the mesh model should be similar to the one before deformation.

Constraint D. Stretching and bending constraint. The stretching and bending of the mesh model should follow the elastic properties of tissue. They are should be minimized in most cases to avoid stenosis.

Constraint E. Smooth join constraint. In some cases, the join should be smooth after deformation and joining.

Among these above constraints, Constraints A and B are hard constraints that must be satisfied. Constraints C, D and D are soft constraints that may contradict each other. So it might not be possible to exactly satisfy all the soft constraints. Instead, a balance between these constraints should be achieved depending on the surgical requirements.

4.2.3 High-level Requirements

High-level surgical procedure corresponds to a high-level computational procedure that comprises a sequence of mid-level computational tasks and low-level mesh operations. For example, the simulation of ASO consists of the following computational tasks:

1. Transection of aorta and pulmonary artery.
2. Transfer of coronary artery.
3. Reconstruction of neo-aorta.
4. Reconstruction of neo-pulmonary artery.

Similarly, the simulation of Norwood operation consists of the following computational tasks:

1. Transection of aorta and pulmonary artery (similar to ASO).
2. Closure of pulmonary artery.
3. Reconstruction of neo-aorta (similar to ASO).
4. Placement of a tube graft to connect aorta to pulmonary artery.

Besides simulating surgical procedures, the system should also provide supporting functions to assist the surgeons in using the system:

1. Size measurement:
   The size of a patch must be accurately measured in real units (e.g., cm or mm) because the surgeon needs to know the exact size of the patch to be cut.
2. Visualization aid:
   The system should visualize important aspects of the mesh models such as the amount of stretching and twisting of the models after deformation. This allows the surgeons to assess whether the amount of deformation is acceptable. To show the joins, different mesh models should be rendered with different colors.

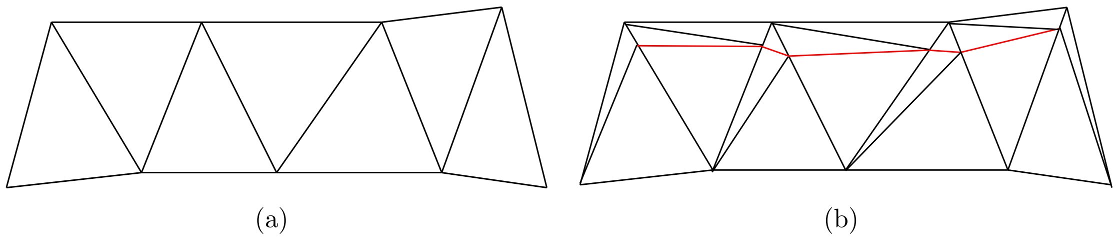


Figure 4.8: Subdivision of mesh. (a) A triangle mesh. (b) A cut in the mesh produced by subdivision method (indicated by the red polyline). Some narrow triangles are generated.

4.3 Candidate Algorithms

The low-level mesh operations and mi-level computational tasks need to be supported by appropriate algorithms. High-level procedures just comprise low-level and mid-level algorithms. So, this section discusses cardiac algorithms for low-level mesh operations and mid-level computational tasks.

4.3.1 Low-level Algorithms

Each low-level mesh operation is supported by a mesh manipulation algorithm. For mesh cutting, direct subdivision method [14, 25] (Figure 4.8) can be used. It creates accurate cut that changes the topology of the mesh model. New vertices and edges are introduced along the cut path, which may result in narrow triangles (Figure 4.8(b)) that could affect further manipulation of the mesh. These narrow triangles should be avoided. Delaunay mesh cutting algorithm [42, 43] may also be used for the purpose of avoiding new generated narrow triangles after cutting.

For mesh deformation, the Differential Geometry (DG) method can be adopted. As discussed in Section 3.2, Laplacian model [32, 34, 57] is suitable and efficient for simulation of soft tissue deformation. The mesh deformation algorithm should impose various constraints derived from surgical requirements. The original Laplacian method already handles Constraints A and C. To impose Constraints D and E, the Laplacian method needs to be extended (see Chapter 5 for details). Imposing constraint B (no penetration constraint) into this deformation scheme is a challenging task. Collision detection techniques [32, 34, 57] may be used.

Poisson method [46, 73, 74], another DG method, is a candidate algorithm for mesh deformation of patches in smooth joining operation. It models the continuity and smoothness at the joins for mesh deformation.

For mesh joining, simple polyline merging method [14, 25] be used to merge the meshes that are joined at their boundaries. Note that the joining boundaries of the meshes are discrete, i.e., they are polylines. There may not be a one-to-one mapping between the two vertex sets of the joining boundaries. Some new vertices may need to be introduced to make sure that the join is leakage-free.

A summary of the low-level operations and candidate algorithms is given in Table 4.1.

Table 4.1: Low-level basic surgical operations, mesh operations and their corresponding candidate algorithms.

Basic surgical operations	Mesh operations	Candidate algorithms
Cutting	Cutting	Direct subdivision method [14, 25], Delaunay mesh cutting [42, 43]
Displacement	Displacement + Deformation	Laplacian method [32, 34, 57]
Joining	Deformation + Joining	Laplacian method [32, 34, 57], Poisson method [46, 73, 74], Polyline merging method [14, 25]

4.3.2 Mid-level Algorithms

As discussed in Section 4.2.2, a mid-level computational task is formulated as a computational problem. An algorithm needs to be developed to solve the computational problem. The algorithm may involve a sequence of existing algorithms for mesh operations and deformation. The Laplacian or Poisson mesh deformation algorithm needs to be extended to include additional constraints as discussed in Section 4.2.2.

For complex computational problems, novel algorithms will need to be developed. For example, reconstruction of aorta and pulmonary artery are two mid-level tasks that are not independent. Their results affect each other because the reconstructed aorta and pulmonary artery should not penetrate each other and other parts of the heart. These requirement will be solved by complex optimization algorithms. Some examples of these algorithms are presented in Chapter 5.

4.4 Validation

Validation of a surgical simulation system is a very difficult and challenging issue. The ideal method to validate the predicted surgical result is to compare it with the result of a real surgical procedure. However, this approach is not practical. It is not possible to perform 3D scanning of actual surgical results for quantitative comparison. Also, lab testing of human tissues may involve ethical issues.

An alternative approach is to validate the system against a set of objects with a range of elasticity. The idea is to collect a set of objects with varying elasticity, from very rigid to very flexible, with the blood vessels falling in the range. These objects are scanned to obtain their 3D models. Then, some basic surgical operations are performed on the objects and the resulting objects are scanned to obtain ground-truth surgical results. Next, the same surgical operations are simulated on the 3D models of the objects. Finally, the simulation results are compared with the ground-truth surgical results. If the difference between the simulation results and the ground truth is small, then it can be claimed that the system can model blood vessel's elastic properties if the parameters are correctly set.

Table 4.2: Tasks involved in completing the proposed research.

Tasks	Done?
Low-level	
Mesh cutting	yes
Mesh displacement and deformation	partial
Mesh joining	partial
Mid-level	
Transection of AO and PA	yes
Transfer of CA	no
Reconstruction of AO	partial
Reconstruction of PA	no
High-level	
ASO	partial

4.5 Research Schedule

Table 4.2 summarizes the tasks involved in completing the proposed research. Some of the tasks have already been accomplished, and they will be presented in Chapter 5. The remaining tasks will be completed according to the following schedule:

- Present – March 2008: mesh joining, reconstruction of aorta (AO).
- April 2008 – August 2008: mesh deformation with all the constraints, reconstruction of aorta and pulmonary artery (PA).
- September 2008 – November 2008: transfer of coronary arteries (CA).
- December 2008 – June 2009: ASO, validation and thesis writing.

If time permits, Norwood operation and its mid-level computational tasks will also be included.

Chapter 5

Preliminary Work

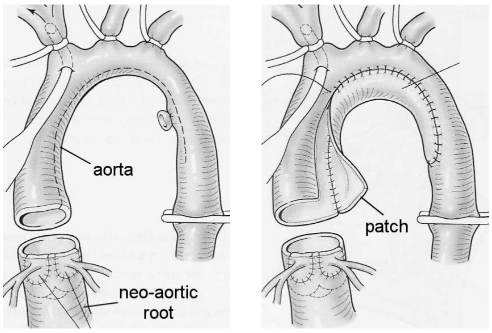
Preliminary work has been done on predictive simulation of neo-aorta reconstruction (Section 5.1) and various cases of smooth joining (Section 5.2). The mesh deformation algorithm developed for this work is based on the Laplacian deformation method [32], imposed with additional constraints: minimal stretching constraint and smooth join constraint.

5.1 Predictive Simulation of Neo-aorta Reconstruction

5.1.1 Problem Definition

In neo-aorta reconstruction, the diameter of the aorta is often smaller than that of the neo-aortic root. So the aorta needs to be cut along its long axis, opened and patched with a additional tissue to enlarge its end [27, 61] (Fig. 5.1). The cutting position and length are typically decided by the surgeon based on anatomical and surgical considerations.

In this work,the aorta and the neo-aortic roots are modeled as tubular (i.e., hollow) 3D meshes with zero surface thickness. That is, only their outer surfaces are modeled. Each model has two ends where surface discontinuities occur. The problem of predictive simulation of aorta reconstruction can be defined as follows:



(a) (b)

Figure 5.1: Reconstruction of aorta [61]. (a) Aorta is cut longitudinally (dashed curve). (b) Aorta is opened and patched to match the diameter of the neo-aortic root.

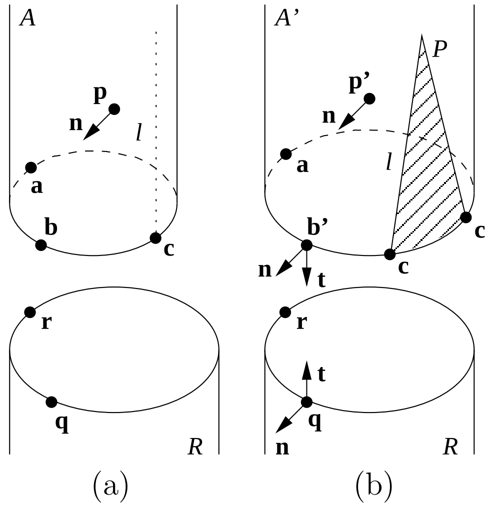


Figure 5.2: Reconstruction of aorta. (a) Aorta A is cut at $\mathbf{c}$ longitudinally (dotted line), (b) opened and patched with a tissue P (shaded surface) to fit neo-aortic root R. Dashed ellipses denote the boundary curves at the ends of A and R, $\mathbf{n}$ is surface normal and $\mathbf{t}$ is surface tangent.

> Given tubular triangulated 2-manifold mesh models (with 1-D boundaries) of the aorta A and the neo-aortic root R, where the diameter of A is smaller than that of R, a cutting position $\mathbf{c}$ at the end of A and a cutting length l along the long axis of A (Fig. 5.2), determine the shapes of the opened aorta A' and the patch P that form smooth, continuous, and leakage-free joins with R.

Simulation of aorta reconstruction requires mesh operations including cutting, deformation, and joining. Mesh cutting is performed using triangle sub-division method. Mesh joining is straightforward. Mesh deformation, on the other hand, is complex. It is applied to open the cut aorta to match the neo-aortic root. The deformation should be subjected to the following constraints to avoid stenosis:

A. The shape of the opened aorta should be similar to that of the original aorta, and its surface should be smooth.

B. The opened aorta should not be stretched or twisted.

C. The opened aorta, neo-aortic root, and the patch should form smooth, continuous, and leakage-free joins.

The predictive simulation problem can be formulated in detail as follows. After cutting A, point $\mathbf{c}$ is split into two points $\mathbf{c}$ and $\mathbf{c}'$, and two new boundary edges are formed in A. Let $\mathbf{a}$ denote the point on the end of A opposite to $\mathbf{c}$. The deformation of A moves a point $\mathbf{p}$ on A to a new position $\mathbf{p}'$ on the opened aorta A', except for one point that locates the global position of A in 3D space, which can be assumed to be $\mathbf{a}$ without loss of generality. So, we shall call point $\mathbf{a}$ the *anchor point*.

To satisfy the shape constraint (Constraint A), the surface normal $\mathbf{n}(\mathbf{p})$ at any point $\mathbf{p}$ on A should be preserved after deformation (Fig. 5.2):

$$\mathbf{n}(\mathbf{p}) = \mathbf{n}(\mathbf{p}'). \tag{5.1}$$

To satisfy Constraint B (minimum stretching and shrinking), the distance d between any two connected mesh points, say, $\mathbf{a}$ and $\mathbf{p}$, should be preserved after deformation, i.e.,

$$d(\mathbf{a}, \mathbf{p}) = d(\mathbf{a}, \mathbf{p}'). \tag{5.2}$$

If the cutting length l is long enough, then A' can be opened wide enough to join with R smoothly. Let $\mathbf{r}$ denote the point on R that coincides with $\mathbf{a}$ when A' and R are joined, i.e., $\mathbf{a} = \mathbf{r}$. Let d_U denote the distance between two boundary points measured along the boundary curve. After joining, any point $\mathbf{b}'$ on the end of A' at a distance $d_U(\mathbf{a}, \mathbf{b}')$ from $\mathbf{a}$ must coincide with point $\mathbf{q}$ on the end of R at the same distance from $\mathbf{r}$, i.e.,

$$\mathbf{a} = \mathbf{r}, \quad \mathbf{b}' = \mathbf{q}, \quad d_U(\mathbf{a}, \mathbf{b}') = d_U(\mathbf{r}, \mathbf{q}). \tag{5.3}$$

One way to satisfy the smooth join constraint (Constraint C) is to ensure that the surface normals at the corresponding end points on A' and R are identical. This method is, however, not robust because the end points are located at surface discontinuities, and their surface normals are, thus, ill-defined. Instead, the surface tangents $\mathbf{t}$ at the end points parallel to the long axes of the arteries are used to specified the constraint (Fig. 5.2):

$$\mathbf{t}(\mathbf{b}') = -\mathbf{t}(\mathbf{q}) \tag{5.4}$$

for each pair of corresponding points $\mathbf{b}'$ on A' and $\mathbf{q}$ on R. The method for estimating the surface tangent will be described in Section 5.1.2.

The patch P should completely cover the opening in A' without leakage. So, it should observe the same continuity conditions across the two joins with A' and the one join with R, as described in Eqs. 5.3 and 5.4, and its shape should be similar to that of A'.

In general, the opened aorta A' may not form a smooth, continuous join with the neo-aortic root R if the minimum stretching constraint is strictly adhered to. By relaxing the constraint, i.e., implementing them as soft constraints, the algorithm can strike a balance between producing a smooth join and minimizing surface stretching.

5.1.2 Predictive Simulation Algorithms

In actual Arterial Switch Operation (ASO), the aorta has to be moved (i.e., translated and rotated) to align with the position and orientation of the neo-aortic root. This movement induces additional deformation on the aorta. Here we wish to study the amount of deformation of the aorta due only to the opening of its end to match the diameter of the neo-aortic root. So, it is assumed that the aorta and the neo-aortic root are already globally aligned in terms of position and orientation. That is, $\mathbf{a} = \mathbf{r}$ and the long axes of A' and R are aligned at the join.

The algorithm for predictive simulation of aorta reconstruction can be summarized into the following steps:

1. Open the cut aorta A:

   (a) Map end points $\mathbf{b}$ on A to new positions $\mathbf{b}'$ on A'.
   (b) Deform the cut A into A', with the new positions $\mathbf{b}'$ as hard constraints, and Eqs. 5.1, 5.2, and 5.4 as soft constraints.

2. Fit the uncut aorta A to A':
   (a) Map end points $\mathbf{b}$ on A to new positions $\mathbf{b}''$ on A''.
   (b) Deform the uncut A into A'' to fit A' and cover the opening, with the new positions $\mathbf{b}''$ as hard constraints and Eq. 5.1 as soft constraint.

3. Compute appropriate patch P:
   (a) Find the points $\mathbf{e}$ on A'' that correspond to the cut edges on A'.
   (b) Cut a patch P' from A'' according to points $\mathbf{e}$.
   (c) Deform P' into P by fitting to the cut edges of A' and the end of R.

Details the algorithm are explained in the following.

Opening Cut Aorta

In Step 1a, each end point $\mathbf{b}$ on A is mapped to the new position $\mathbf{b}'$ using Eqs. 5.2 and 5.3. In real application, the new positions $\mathbf{b}'$ may not correspond to existing mesh points (vertices) on R. To ensure a close fit between A' and R, for each end point $\mathbf{q}$ of R, its corresponding position on A is computed using Eqs. 5.2 and 5.3, and a new mesh point (and associated edges) is added to the boundary curve of the end of A at this position. We shall call these new end points $\mathbf{b}$ as well. Their new positions $\mathbf{b}' = \mathbf{q}$ for a unique $\mathbf{q}$ on R.

Step 1b requires an deformation algorithm. This algorithm is adapted from the differential geometric method described in [32], which handles hard and soft constraints. The hard constraints are positional constraints provided by the positions of points $\mathbf{b}'$, i.e., $\mathbf{b}' = \mathbf{q}$ as represented by:

$$\mathbf{H}\,\mathbf{x} = \mathbf{d} \tag{5.5}$$

where $\mathbf{x}$ is a vector of all the mesh points in A' and $\mathbf{d}$ is a vector of the desired coordinates of $\mathbf{b}'$, and $\mathbf{H}$ is a matrix that relates $\mathbf{x}$ and $\mathbf{d}$.

Shape constraint (Constraint A) is satisfied by minimizing the squared difference (L_2 norm) between the mean-curvature normals before and after deformation, i.e.,

$$\|\mathbf{l}(\mathbf{p}') - \mathbf{l}(\mathbf{p})\| \tag{5.6}$$

The surface normal $\mathbf{n}(\mathbf{p})$ at point $\mathbf{p}$ can be approximated by the Laplacian operator $\mathbf{l}(\mathbf{p})$:

$$\mathbf{l}(\mathbf{p}) = \sum_{\mathbf{v}_i \in N(\mathbf{p})} w_i\,(\mathbf{p} - \mathbf{v}_i) \tag{5.7}$$

where $N(\mathbf{p})$ is the set of neighboring points connected to $\mathbf{p}$ in the mesh. The magnitude of $\mathbf{l}(\mathbf{p})$ approximates the mean curvature at $\mathbf{p}$, and the direction of $\mathbf{l}(\mathbf{p})$ approximates $\mathbf{n}(\mathbf{p})$.

An intuitive understanding of the Laplacian operator is illustrated in Fig. 5.3(a). The Laplacian operator $\mathbf{l}(\mathbf{p})$ sums up the difference vectors $\mathbf{p} - \mathbf{v}_i$ between $\mathbf{p}$ and its connected neighbors $\mathbf{v}_i$. The summation cancels out the tangential components of the difference vectors, and sums up their normal components to produce a vector normal to the surface at $\mathbf{p}$.

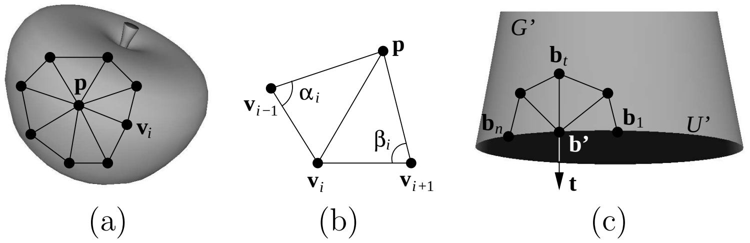


Figure 5.3: Laplacian operator. (a) Non-boundary mesh point $\mathbf{p}$ and its connected neighbors. (b) Definitions of α_i and β_i. (b) Boundary point $\mathbf{b}'$ and its connected neighbors.

For accurate estimation of mean curvature, the so-called *cotangent weights* should be adopted for w_i [32, 57]:

$$w_i = \frac{1}{2A(\mathbf{p})}(\cot \alpha_i + \cot \beta_i) \tag{5.8}$$

where $A(\mathbf{p})$ is the Voronoi area of point $\mathbf{p}$, and α_i and β_i are the angles opposite to the edge connecting $\mathbf{p}$ and $\mathbf{v}_i$ (Fig. 5.3(b)). On the other hand, for simplicity, uniform weights may be used also, i.e., $w_i = 1/|N(\mathbf{p})|$, where $|N(\mathbf{p})|$ is the number of neighboring points connected to $\mathbf{p}$.

As the cut edges of A' are surface discontinuities, the Laplacians of the edge points do not correspond to surface normals. So, a different constraint is imposed to constrain the shape of the cut edges. Let $\mathbf{b}'$ denote an edge point on A', and $\mathbf{b}_1$ and $\mathbf{b}_n$ denote two other edge points on A' that are connected to $\mathbf{b}'$. Then, minimizing the term

$$\|(\mathbf{b}' - \mathbf{b}_1) + (\mathbf{b}' - \mathbf{b}_n)\| \tag{5.9}$$

will straighten the cut edges of A'. This is because the distance between the three points are preserved after deformation by the geodesic distance constraint. Therefore, the two vectors $\mathbf{b}' - \mathbf{b}_1$ and $\mathbf{b}' - \mathbf{b}_n$ would be close to parallel.

The constraints given in Eq. 5.6 and 5.9 are assembled into the shape change energy E_s to be minimized:

$$E_s = \|\mathbf{L}\,\mathbf{x} - \mathbf{u}\|. \tag{5.10}$$

The term $\mathbf{L}\,\mathbf{x}$ contains $2\mathbf{p}' - \mathbf{b}_1 - \mathbf{b}_n$ for edge points $\mathbf{p}'$ and $\mathbf{l}(\mathbf{p}')$ for non-edge points $\mathbf{p}'$. The term $\mathbf{u}$ contains the value 0 for edge points $\mathbf{b}'$ and $\mathbf{l}(\mathbf{p})$ for non-edge points $\mathbf{p}'$.

Note that there are two ways to compute E_s for non-boundary points: using either uniform weights or cotangent weights (Eq. 5.8) for the Laplacian operators. For neo-aorta reconstruction, we adapt uniform weights. The difference between these two implementations is examined in Section 5.2, where the reason for choosing uniform weights is given.

Minimum stretching (Constraint B) is achieved by constraining the distances between every pair of connected neighboring points. For every edge e_{ij} that connects point $\mathbf{p}_i$ and $\mathbf{p}_j$ in A, the stretching energy is formulated as

$$k_{ij}\left(\|\mathbf{p}'_i - \mathbf{p}'_j\| - \|\mathbf{p}_i - \mathbf{p}_j\|\right)^2 \tag{5.11}$$

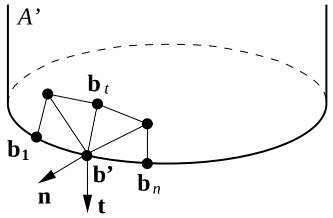


Figure 5.4: End points and their connected neighbors of the curved surface of A'.

where k_{ij} is the stiffness coefficient of e_{ij}. This is a non-linear constraint because of the Euclidean distances. It can be re-written as

$$k_{ij}\,\|\mathbf{d}_{ij}-\mathbf{s}_{ij}\|, \tag{5.12}$$

where

$$\begin{aligned}\mathbf{d}_{ij} &= \mathbf{p}'_i-\mathbf{p}'_j\\ \mathbf{s}_{ij} &= \mathbf{d}_{ij}\frac{\|\mathbf{p}_i-\mathbf{p}_j\|}{\|\mathbf{p}'_i-\mathbf{p}'_j\|}\end{aligned} \tag{5.13}$$

Assembling the edge stretching energy together gives the total stretching energy E_g to be minimized:

$$E_g = \|\mathbf{D}\,\mathbf{x}-\mathbf{s}(\mathbf{x})\| \tag{5.14}$$

where $\mathbf{D}\,\mathbf{x}$ contains $\mathbf{d}_{ij}$ terms and $\mathbf{s}(\mathbf{x})$ contains $\mathbf{s}_{ij}$ terms.

One approach to satisfying the smooth join constraint (Constraint C) is to ensure that the surface normals at the corresponding points on G' and H are identical. Unfortunately, the Laplacian operator at a boundary point does not correctly estimate the mean-curvature normal. As illustrated in Fig. 5.3(c), the summation of the Laplacian operator does not cancel out the tangential components of the difference vectors between $\mathbf{b}'$ and its connected neighbors.

Our method of imposing Constraint C is achieved by matching the surface tangents of the corresponding end points $\mathbf{b}'$ on A' and $\mathbf{q}$ on R. As there is more than one surface tangent at a point, the one that is normal to the boundary curve is chosen.

First, we start with the original end point $\mathbf{b}$ on A. Let $\mathbf{b}_1,\ldots,\mathbf{b}_n$ denote the connected neighboring points of $\mathbf{b}$ such that $\mathbf{b}_1$ is also an end point as shown in Fig. 5.4. Among these connected neighbors, there is a neighbor $\mathbf{b}_t$ such that $\mathbf{b}-\mathbf{b}_t$ is (approximately) normal to the surface and parallel to the long axis of A'. We call this point the *tangential point*.

The tangential point $\mathbf{b}_t$ of $\mathbf{b}$ is computed as follows. First, the normal $\mathbf{n}(\mathbf{b})$ at $\mathbf{b}$ is estimated by the weighted mean of the normals of the neighboring faces of $\mathbf{b}$. Then, the surface tangent $\mathbf{t}(\mathbf{b})$ is computed as the cross-product $\mathbf{n}(\mathbf{b})\times(\mathbf{b}-\mathbf{b}_1)$, where $\mathbf{b}_1$ is the end point connected to $\mathbf{b}$. Finally, the tangential point $\mathbf{b}_t$ of $\mathbf{b}$ is the point whose difference vector $\mathbf{b}-\mathbf{b}_t$ is most parallel to $\mathbf{t}(\mathbf{b})$. This tangential point is used as an approximation of the tangential point of $\mathbf{b}'$, whose position is to be computed by the deformation algorithm. The difference vector $\mathbf{b}'-\mathbf{b}_t$ is used to express the smooth join constraint.

The desired surface tangent at $\mathbf{b}'$ is given by the tangent $\mathbf{t}(\mathbf{q})$ at the corresponding end point $\mathbf{q}$ on R. This tangent $\mathbf{t}(\mathbf{q})$ is computed in the same way as described above. Now, the constraint can be specified as

$$\mathbf{b}' - \mathbf{b}_t = -\mathbf{t}(\mathbf{q})\frac{\|\mathbf{b} - \mathbf{b}_t\|}{\|\mathbf{t}(\mathbf{q})\|}. \tag{5.15}$$

Equation 5.15 can be assembled to formulate the continuous join energy:

$$E_j = \|\mathbf{F}\mathbf{x} - \mathbf{w}\| \tag{5.16}$$

where $\mathbf{w}$ contains the known terms in Eq. 5.15, and $\mathbf{F}$ relates $\mathbf{x}$ and $\mathbf{w}$. The total energy E to be minimized is the weighted sum:

$$E = E_s + \lambda_g\, E_g + \lambda_j\, E_j \tag{5.17}$$

which is equivalent to minimizing

$$\|\mathbf{K}\,\mathbf{x} - \mathbf{z}(\mathbf{x})\| \tag{5.18}$$

subject to the hard constraint Eq. 5.5, where $\mathbf{K} = (\mathbf{L}^T\ \mathbf{D}^T\ \mathbf{F}^T)^T$ and $\mathbf{z}(\mathbf{x}) = (\mathbf{u}^T\ (\mathbf{s}(\mathbf{x}))^T\ \mathbf{w}^T)^T$. This is a non-linear optimization problem with equality constraints. It can be solved iteratively using Gauss-Newton method, as demonstrated in [68], with the equality-constrained least squares method described in [32]. The details of the solving procedure are described in Appendix A.

In the current implementation, $\lambda_j = 20$, $\lambda_g = 20$ for boundary points, and $\lambda_g = 0.1$ for non-boundary points. λ_g for non-boundary points is set to a small value to allow for some small amount of stretching. If it is set to too large a value, the model will fold instead of stretch in order to preserve geodesic distances.

Computing Appropriate Patch

After opening aorta A into A' and joining it to neo-aortic root R, the next step is to determine an appropriate patch P to fill up the opening or hole. One possible method is to apply shape completion algorithms such as [46]. These algorithms find an appropriate patch on the object either manually or automatically based on shape features, and then deform it to fit the hole. In our application, this patch cannot be found on the original aorta A because A cannot fit R. It may not be found on A' because, after deformation, the shape of A' is not uniform. Instead, the patch is determined using a surface fitting approach in our algorithm.

The main idea of the approach is to deform the original uncut aorta A using our algorithm into A'' to fit A' and cover the opening (Fig. 5.5). Then, the patch P can be determined from the fitted model A''.

To perform the surface fitting, the correspondence between the points on A and A'' is first established (Step 2a). The end points $\mathbf{b}$ on A are mapped to their new positions $\mathbf{b}''$ on A'', which correspond to some end points $\mathbf{q}$ on R, as follows. The perimeters γ of the ends of A and R are measured, and a scaling factor k is computed as the ratio of $\gamma(A)$ over $\gamma(R)$. The anchor points on A, A'', and R are the same point, as described in Section 5.1.2: $\mathbf{a} = \mathbf{r}$. An end point $\mathbf{b}$ on A is mapped to the end of $\mathbf{q}$ on R at the corresponding distance from the anchor point: $g(\mathbf{a}, \mathbf{b}) = k\, g(\mathbf{r}, \mathbf{q})$. So, the new position $\mathbf{b}'' = \mathbf{q}$. There is no a prior information about which mesh points on A should mapped to the edge points on A''. So, this mapping is omitted at this stage. After mapping, A is deformed with the end points' new positions $\mathbf{b}''$ as hard constraints, and the shape constraint and smooth join constraint as software constraints (Step 2b).

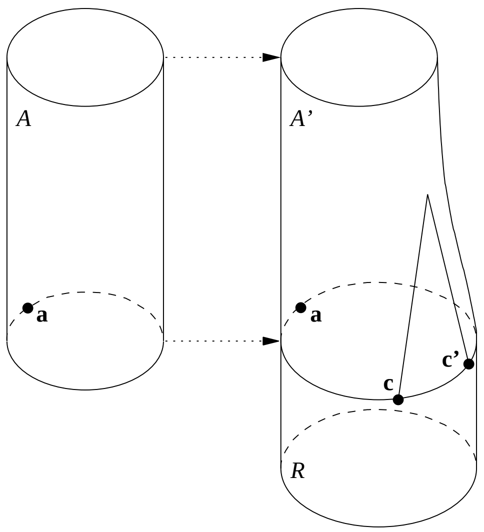


Figure 5.5: Deform original uncut A to fit A'. Dotted arrows denote the mapping of end points on A to A' and R.

The deformed A'' will have a similar shape as A and the end points of A' and R will fall on the surface of A''. On the other hand, the points of the cut edges of A' may not fall on the surface of A' because no constraint is specified for them as yet. After fitting, for each edge point $\mathbf{e}'$ on A', its surface normal is estimated by the weighted average of the normals of its neighboring faces. Then, a ray is cast along the normal direction. The intersection of the ray and the surface of A'' is regarded as the corresponding point $\mathbf{e}$ on A'' (Step 3a).

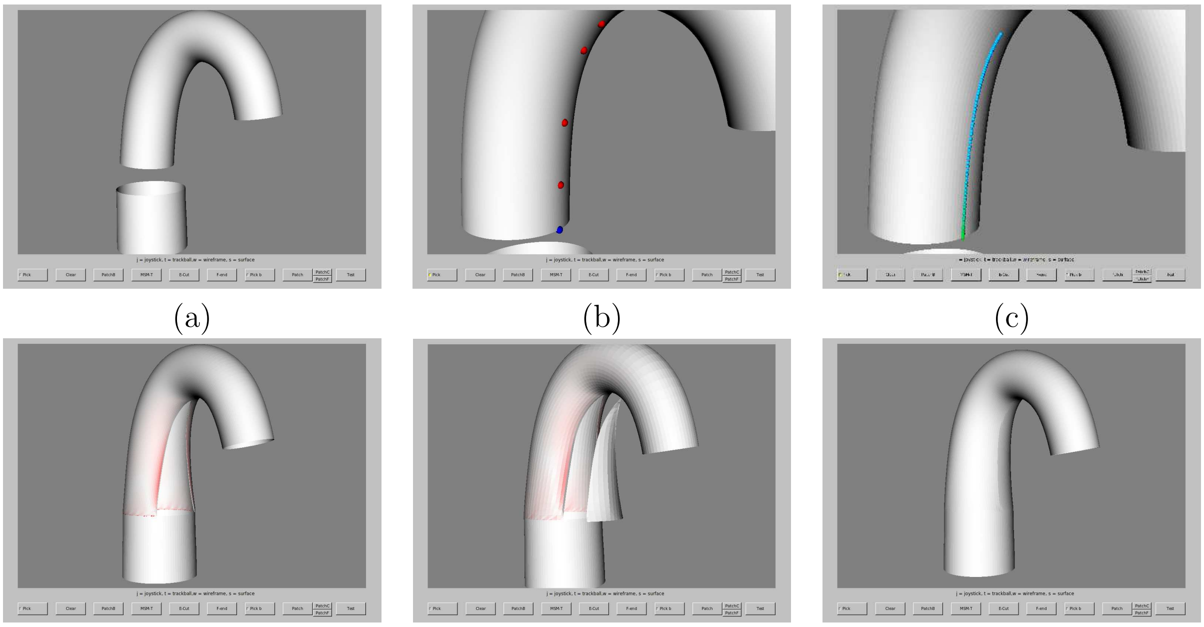
Next, a patch P' on A' is cut according to the points $\mathbf{e}$ on A'' (Section 3b). Finally, the patch P' is deformed into the desired patch P by fitting to the opening of A', with the edge point correspondence as the hard constraint, and shape constraint and smooth join constraint as soft constraints. This step completes the algorithm for predictive simulation of aorta reconstruction.

5.1.3 Experimental Results

Experiments were performed to evaluate the performance of predictive simulation algorithm. For the experiments, 3D mesh models of the aorta and neo-aortic root were created manually based on actual anatomical models.

Figure 5.6 illustrates an example sequence of steps in predictive simulation of aorta reconstruction. First, the cutting point and cutting path are indicated by the user, e.g., a surgeon (Fig. 5.6(b)), by means of several dots on the cutting path. Then, the system automatically computes a cutting path to cut the aorta (Fig. 5.6(c)). Next, the cut aorta is opened using the geodesic distance preserving deformation, and joined to the neo-aortic root (Fig. 5.6(d)). Finally, the patch is computed and fitted to the opening to complete the aorta reconstruction (Fig. 5.6(e, f)).

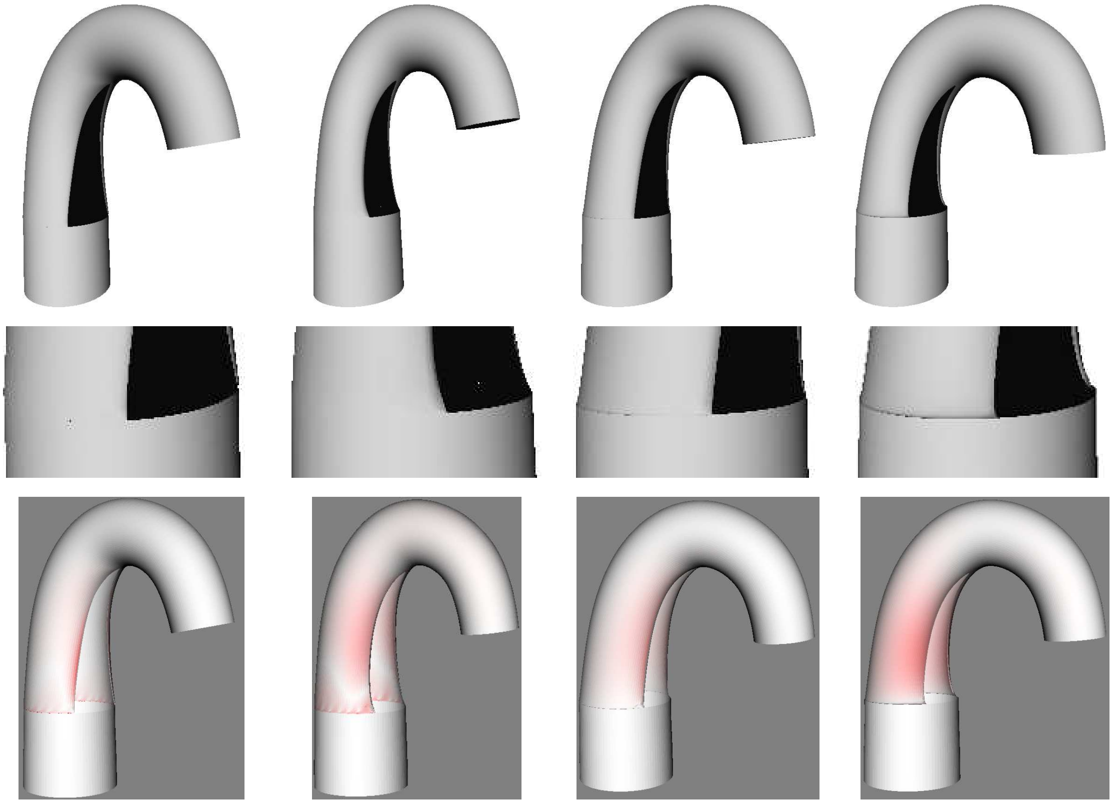
To evaluate the effectiveness of the minimum stretching constraint and smooth join constraint, the deformation algorithm for opening the aorta was compared with three other versions that partially satisfy the constraints (Fig. 5.7). Stretching of the mesh surface is computed by measuring the amount of change in the areas of the triangles before and after



(a) (b) (c)

(d) (e) (f)

Figure 5.6: Demonstration of surgical simulation of aorta reconstruction. (a) Initial configuration. (b) Blue dot indicates cutting point, and red dots indicate cutting path. (c) Blue curve indicates the cutting path computed by the system. (d) Aorta is opened and joined to the neo-aortic root. Red color denotes slight amount of stretching required at the edges. (e) The patch is computed, and (f) joined to the aorta and neo-aortic root.



(a) $G \wedge J$ (b) $\neg\, G \wedge J$ (c) $G \wedge \neg\, J$ (d) $\neg\, G \wedge \neg\, J$.

Figure 5.7: Comparison of deformation algorithms. (a) The optimal result satisfies both (G) minimum stretching constraint and (J) smooth join constraint. (b–d) Results that partially satisfy the constraints. Second row shows enlarged view of the join. Red color in the third row indicates the amount of stretching after deformation (white: no change, light red: small change).

deformation. Test results show that the amount of stretching on the opened aorta is minimal when the minimum stretching constraint is satisfied (Fig. 5.7(a), (c)). On the other hand, when the minimum stretching constraint is not satisfied, there are significant stretching of the opened aorta near the join and the cut edges (Fig. 5.7(b), (d)). Imposing the smooth join constraint produces smooth, continuous joins between the opened aorta and the neo-aortic root (Fig. 5.7(a, b)). Without the smooth join constraint, sharp edges are formed at the join between the aorta and neo-aortic root (Fig. 5.7(c, d)).

Figure 5.8 compares the deformation that satisfies all the constraints (the optimal result) with the one that satisfies only positional hard constraint and minimum stretching soft constraint. Test results show that without shape constraint and smooth join constraint, the algorithm attempts to avoid stretching in extreme (Fig. 5.8(b, d)). Therefore, there is no significant stretching of the mesh surface but the model deforms like a piece of paper even though the shape of the model is more or less preserved.

Although nonlinear least square solver is used to compute the deformation, the execution time is satisfactory. Figure 5.9 shows the convergence of the algorithm. The vertical axis

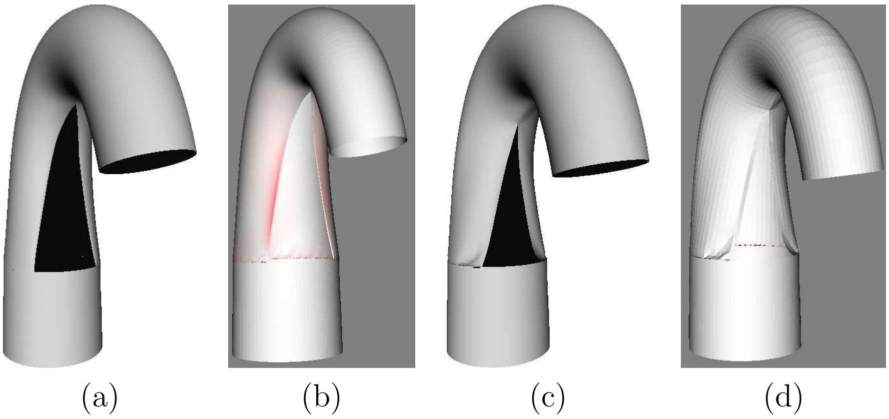


Figure 5.8: Comparison of deformation algorithms. (a) The optimal result satisfies all constraints. (b) The result that satisfies only positional and geodesic distance constraints.

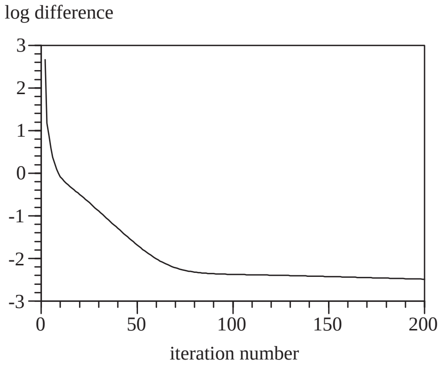
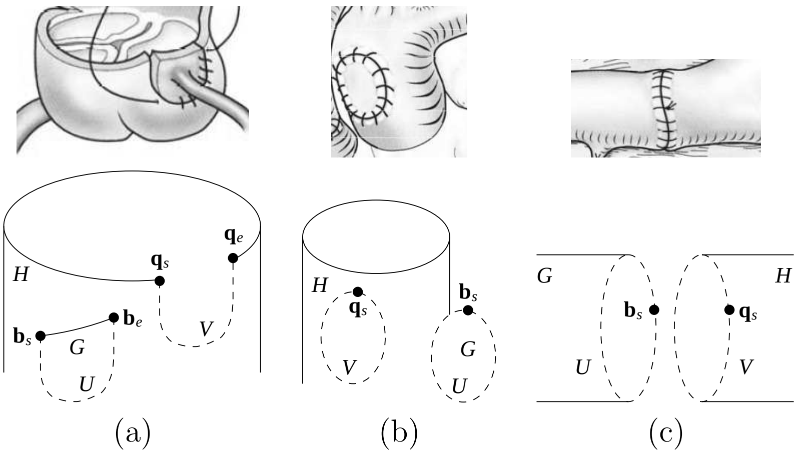


Figure 5.9: Convergence curve of iterative nonlinear solver.

shows the amount of changes between successive iterations, which is measured by the logarithm of the mean-squared difference between the positions of the mesh points at successive iterations. The test result shows that the algorithm began to converge to a stable solution at about 100 iterations even though it was terminated after running for 200 iterations. In an Intel Pentium 4 3GHz CPU with 1GB RAM., the algorithm took about 2 minutes to solve the deformation problem of a model with 2,500 points and 13,000 soft constraints. For a deformation with 10,000 points and 50,000 soft constraints, the algorithm took about 15 minutes to solve.

5.2 Simulation of Smooth Joins

The above deformation algorithm has also been applied on smooth joining of mesh models for simulation of suturing or gluing of arteries. Suturing or gluing of cardiac tissues to form smooth joins are common operations in cardiac surgery. Smooth joining is a general case of the joining operation in opening cut aorta (Section 5.1). So the deformation algorithm with



(a) (b) (c)

Figure 5.10: Various kinds of smooth joins in cardiac surgery. (a) Partial patching. (b) Complete patching. (c) End-to-end join. (Top row) Schematic drawings from [20]. Joins are indicated by the stitch marks. (Bottom row) Computational models. Dashed curves denote corresponding boundaries to be joined.

similar constraints can be used here.

Figure 5.10 illustrates several typical scenarios, including the suturing of coronary button to a slot cut out of the aorta (Fig. 5.10(a)), suturing of a patch of tissue to completely cover a hole on the arterial wall (Fig. 5.10(b)), and end-to-end joining of two arteries (Fig. 5.10(c)).

The two anatomical parts to be joined are modeled as 3D meshes with zero surface thickness. One of the two parts, typically the larger one, is modeled as a fixed, rigid object called the host H. The other one is modeled as a flexible object called the guest G that is deformed to fit the host to form a smooth and continuous join. The general case in which both the host and the guest are flexible and deformable will be investigated in future work. The boundary curves of G and H to be joined, denoted as U and V , are identified by corresponding starting points $\mathbf{b}_s$ and $\mathbf{q}_s$, and end points $\mathbf{b}_e$ and $\mathbf{q}_e$ (Figure 5.10). For closed curves, the starting points are also the end points.

The curves U and V should have identical shape and length. In surgical practice, this can be achieved by manually trimming the tissues to fit each other. The accuracy of this operation depends on the surgeons skill and experience. In our system, we formulate a more general problem in which the shape and length of U and V may differ. The host is fixed and the guest is allowed to stretch or shrink. The minimum amount of deformation required to produce a smooth join is computed and displayed. In this way, a surgeon can assess whether the amount of deformation is acceptable empirically. In reality, cardiac tissues can stretch and fold but cannot shrink. The case in which both the guest and the host can stretch but not shrink will be investigated in future work.

Now, the joining problem can be defined as follows:

> Given two triangulated 2-manifold mesh models with 1-D boundaries, of which one is a rigid host H and the other a flexible guest G, and their corresponding boundary curves V and U to be joined, determine the shape of the deformed guest that forms smooth, continuous, and seamless join with the host. That is,

the shape and length of the deformed U, denoted as U', should be identical to those of V.

The deformation of the guest model G is subjected to the same three constraints as described in Section 5.1.

This section evaluates the effectiveness and efficiency of the deformation algorithm in producing smooth joins of mesh models. Three test scenarios were constructed manually based on the application examples illustrated in Figure 5.10.

The first experiment evaluates the joins produced by the deformation algorithm. For comparison, four variants of the algorithm were tested: (1) For shape constraints, the Laplacian operators adopted either cotangent or uniform weights. (2) For smooth join constraints, either the Laplacian operators or the surface tangents of the boundary points were matched. The test scenario of end-to-end join was used. Three test samples were constructed, with the length of the guest's boundary smaller than, equal to, and greater than that of the host's boundary.

Test results show that matching of Laplacian operators of boundary points could not produce smooth joins (Figure 5.11(2, 3)) because Laplacian operators of boundary points are ill-defined. On the other hand, matching surface tangents successfully produced smooth joins (Figure 5.11(4, 5)) because surface tangents of boundary points can be matched accurately.

In the tests, the right end of the guest model was joined to the host model. The remainder of the guest model was free to move while keeping its shape. When cotangent weights were used in the Laplacian operators for shape constraints, the guest was displaced significantly (Figure 5.11(2, 4)) in order to produce a smooth join with minimum deformation. On the other hand, when uniform weights were used, the guest model was not displaced so significantly (Figure 5.11(3, 5)). This is because the cotangent weights are dependent on the Voronoi areas and the cotangents of the triangles, and they can differ significantly between triangles. As a result, it is difficult to set the weighting parameter λ_s in a manner that properly balances the various constraints. Uniform weights, on the other hand, does not have this problem. Figure 5.11(5) shows that, with uniform weights for shape constraints and matching of surface tangents for smooth join constraints, the algorithm can produce smooth joins successfully regardless of the length of the guest's boundary relative to that of the host's boundary.

The second experiment evaluates the effectiveness of the proposed algorithm in producing a variety of smooth joins. Uniform weights were used for the Laplacian operators for shape constraints. Two variants of the algorithm was tested for comparison: with and without smooth join constraints. All three test scenarios were used. Three test samples were constructed for each scenario, with the length of the guest's boundary smaller than, equal to, and greater than that of the host's boundary. For partial and complete patching, the guest models were flat surfaces, which agree with real applications. The amount of deformation (stretching or shrinking) of the guest model was computed by measuring the amount of change in the areas of the triangles before and after deformation.

Figures 5.12–5.14 show that the smooth join constraint is necessary for the algorithm to produce a smooth join between the guest and the host. The constraint ensures that the algorithm achieves a balance between smooth joining and preservation of the orientation and shape of the guest model.

When the guest' boundary was shorter than the host' boundary, the guest was stretched significantly to match the host (column (a)). When they had the same lengths, the guest was

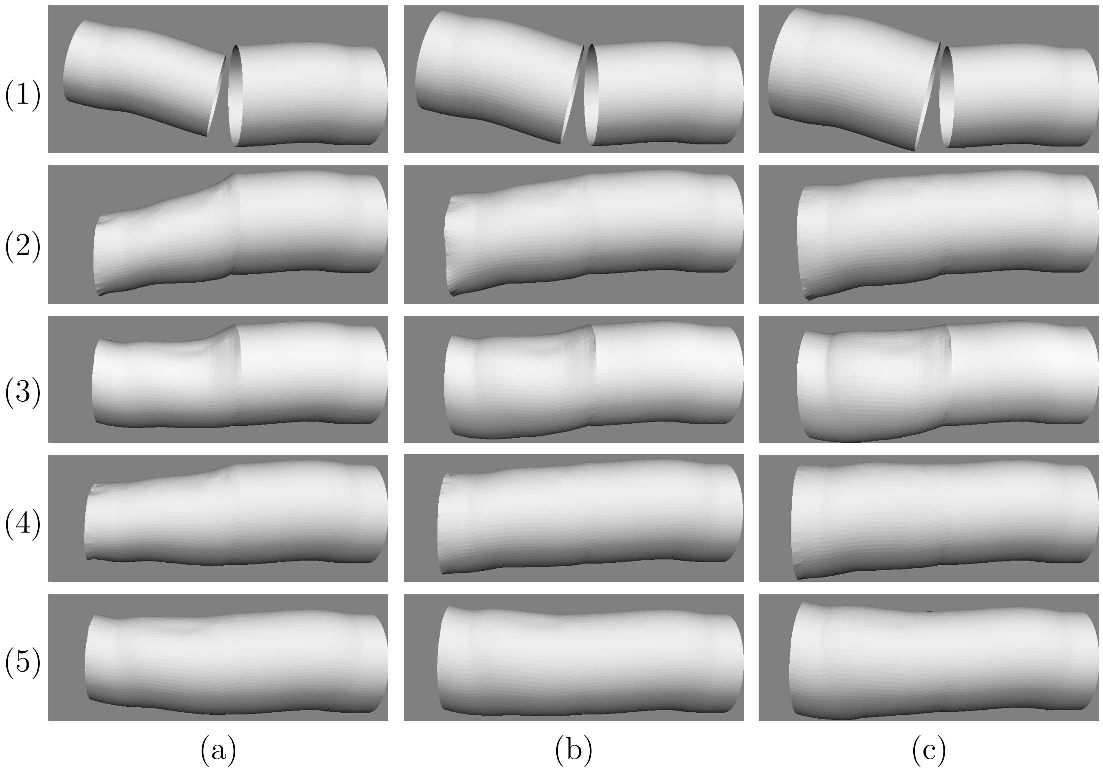


Figure 5.11: Comparison of smooth joining methods. (1) Initial configuration. (2, 3) Matching Laplacian operators. (4, 5) Matching surface tangents. (2, 4) Cotangent weights. (3, 5) Uniform weights. The guest is on the left of the host. The guest's boundary is (a) shorter than, (b) equal to, and (c) longer than the host's boundary.

minimally deformed (column (b)). When the guest's boundary was longer than the host's boundary (column (c)), the guest did not shrink significantly due to the minimum shrinking constraint. Instead, it bulged to match the shape of the host. This is an expected result in real applications.

The computation time of the algorithm was measured in the same PC. The algorithm took 70 seconds to solve the end-to-end join scenario (Figure 5.12(4)), which was an optimization problem with 2,494 points, 522 hard constraints, and 29,688 soft constraints. For partial patching (Figure 5.13(4)), with 1509 points, 1,218 hard constraints, and 17,967 soft constraints, the algorithm took 13 seconds to solve. For complete patching (Figure 5.14(4)), with 571 points, 798 hard constraints, and 6,843 soft constraints, the algorithm took only 2 seconds to solve. The computation can be further accelerated through using more efficient sparse linear solvers [63].

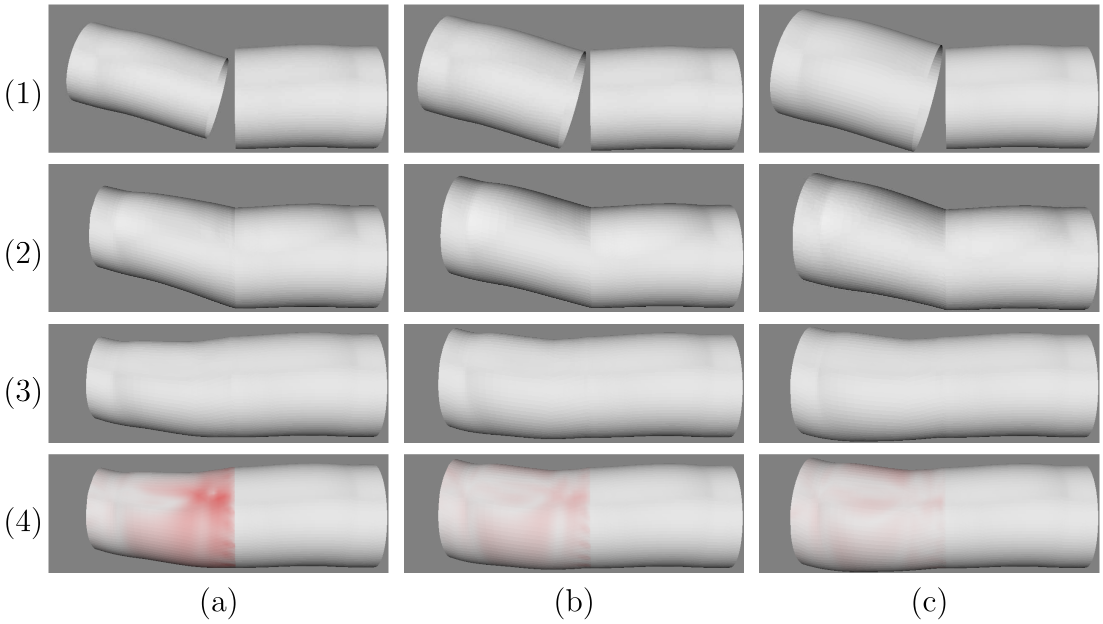


Figure 5.12: Results of end-to-end joint. (1) Initial configurations. (2) Without smooth join constraint. (3, 4) With smooth join constraint. The guest's boundary is (a) shorter than, (b) equal to, and (c) longer than the host's boundary. Red color indicates the amount of deformation of the guest model (white: no change, dark red: large change).

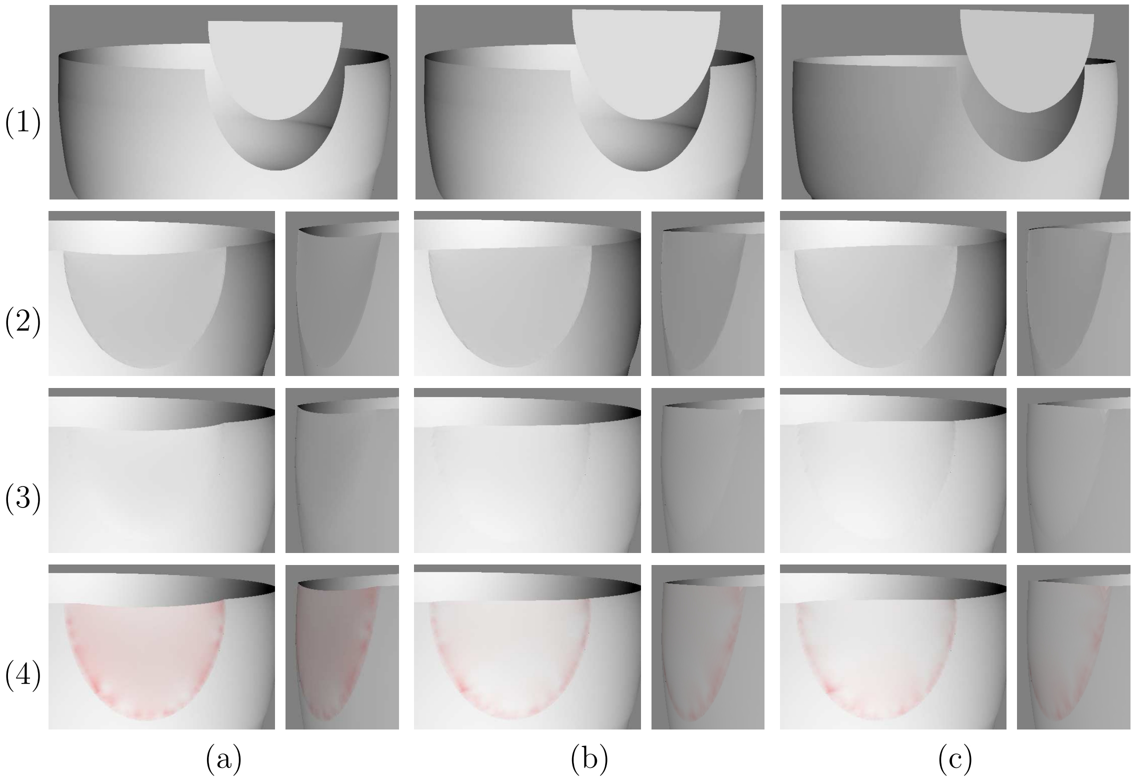


Figure 5.13: Results of partial patching. (1) Initial configurations. (2) Without smooth join constraint. (3, 4) With smooth join constraint. The guest's boundary is (a) shorter than, (b) equal to, and (c) longer than the host's boundary. Red color indicates the amount of deformation of the guest model (white: no change, dark red: large change).

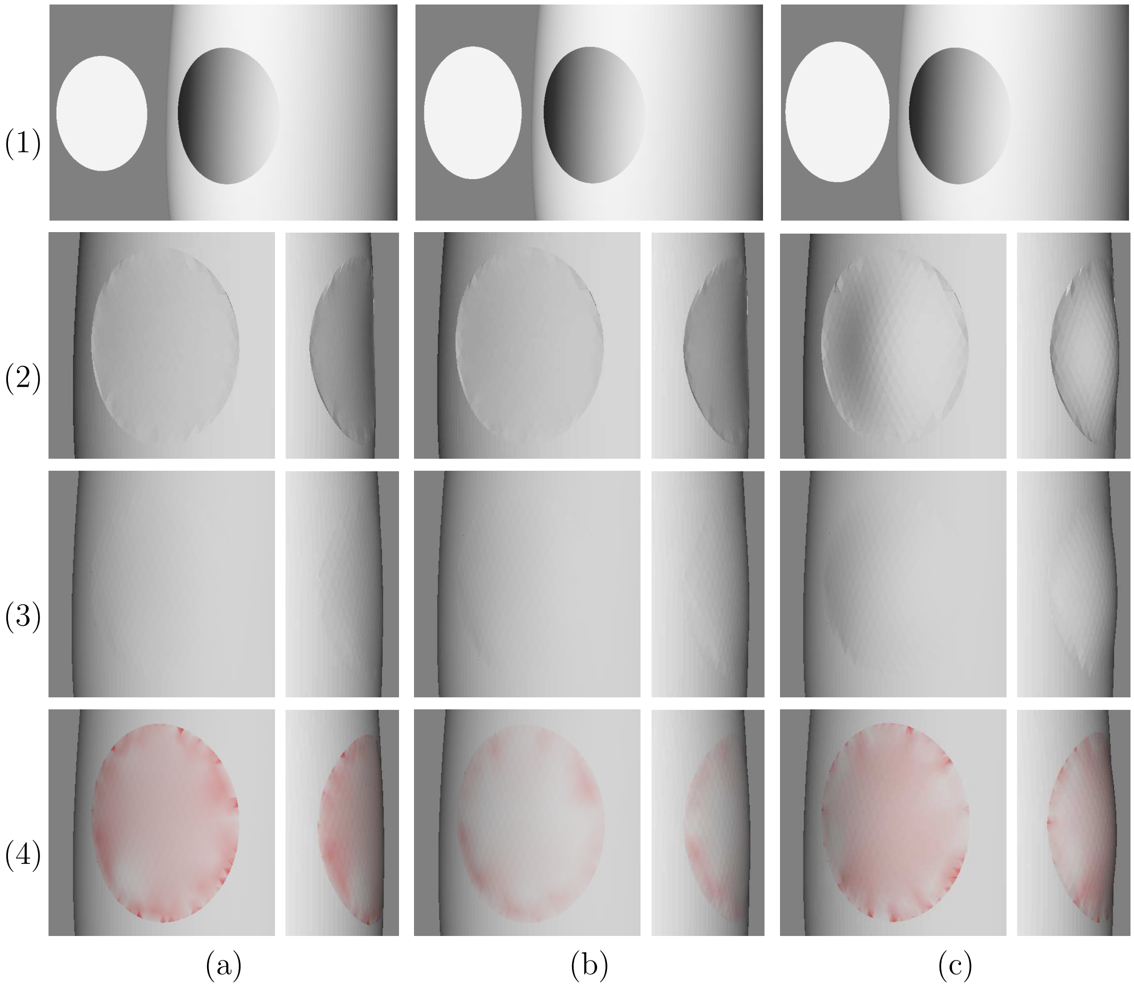


Figure 5.14: Results of complete patching. (1) Initial configurations. (2) Without smooth join constraint. (3, 4) With smooth join constraint. The guest's boundary is (a) shorter than, (b) equal to, and (c) longer than the host's boundary. Red color indicates the amount of deformation of the guest model (white: no change, dark red: large change).

Chapter 6

Conclusion

Many cardiac surgeries involve very complex operations on the heart, the great arteries and other cardiac tissues. At present, cardiac surgeons rely only on echocardiography, cardiac catherization and CT images to understand the specific anatomical structures of a patient. A predictive simulation system is needed for them to perform precise and effective surgical planning of cardiac surgeries.

The literature review shows that there are few surgical simulation systems developed for open surgeries. In particular, existing systems for cardiac surgery perform only simple surgical operations for the repair of ventricular septal defect. Therefore, predictive simulation of complex cardiac surgeries such as ASO and Norwood procedure is an important and challenging research topic.

In this thesis proposal, a predictive simulation system is proposed for complex cardiac surgeries. Surgical requirements of the system are analyzed from the surgeon's perspective. These requirements are grouped into three levels: (1) low-level basic surgical operations, (2) mid-level surgical tasks and (3) high-level surgical procedures. They are translated into system requirements that define the surgical simulation system. The system requirements are also grouped into three corresponding levels: (1) low-level mesh operations, (2) mid-level computational tasks and (3) high-level computational procedures.

Candidate algorithms for achieving these system requirements are discussed. The low-level requirements can be achieved using existing mesh operation algorithms. The mid-level requirements, on the other hand, need to be formulated as equivalent computational problems. Novel algorithms need to be developed for solving these computational problems.

With careful analysis of the system requirements and formulation of the computational problems involved in the simulation of cardiac surgeries, a list of research tasks are identified for completing the proposed research. Some of these tasks have already been accomplished and are presented as preliminary work. The preliminary work shows that the approach adopted in this proposal is feasible and promising for developing a predictive simulation system for complex cardiac surgeries.

Appendix A

Solving Process of DG method

The total energy to be minimized is

$$E = E_s + \lambda_g E_g + \lambda_j E_j, \tag{A.1}$$

subject to the hard constraint Eq. 5.5, which is equivalent to

$$\min_{\mathbf{Hx}=\mathbf{d}} \|\mathbf{A}\,\mathbf{x} - \mathbf{c}(\mathbf{x})\|_2 \tag{A.2}$$

where

$$\mathbf{A} = \begin{pmatrix} \mathbf{L} \\ \mathbf{F} \\ \mathbf{D} \end{pmatrix}, \qquad \mathbf{c}(\mathbf{x}) = \begin{pmatrix} \mathbf{u} \\ \mathbf{w} \\ \mathbf{s}(\mathbf{x}) \end{pmatrix} \tag{A.3}$$

This is a non-linear optimization problem with equality constraints. Since the matrix **A** does not change, the system can be solved iteratively using Gauss-Newton method, collaborated with the equality-constrained least squares method described in [32].

The first step is to incorporate the hard constraints into the optimization scheme. Since our hard constraints are only applied on the vertices, and each vertex has at most one hard constraint. The matrix $\mathbf{H} \in R^{m\times n}$ has full rank. Each row of **H** is a vector that has only one 1, other positions are 0.

So $\mathbf{H}^T$ can be simply decomposed as

$$\mathbf{H}^T = \mathbf{QR} \tag{A.4}$$

where

$$\begin{aligned} \mathbf{Q} &= \begin{bmatrix} \mathbf{H}^T \ \mathbf{Q}_r \end{bmatrix} \\ \mathbf{R} &= \begin{bmatrix} \mathbf{I}_m \\ 0 \end{bmatrix}. \end{aligned} \tag{A.5}$$

Q is a $n \times n$ orthogonal matrix. $\mathbf{Q}_r \in R^{n\times(n-m)}$ contains all the xx vectors that do not appear in $\mathbf{H}^T$. Thus, Eq. 5.5 becomes

$$\mathbf{R}^T\mathbf{Q}^T\mathbf{x} = \mathbf{d}. \tag{A.6}$$

Let

$$\mathbf{Q}^T\mathbf{x} = \begin{bmatrix} \mathbf{u} \\ \mathbf{v} \end{bmatrix}, \tag{A.7}$$

we have

$$\mathbf{u} = \mathbf{H}\mathbf{x} = \mathbf{d}, \tag{A.8}$$

$$\mathbf{x} = \mathbf{Q}\left[\begin{array}{c}\mathbf{u}\\ \mathbf{v}\end{array}\right] = \mathbf{H}^T\mathbf{u} + \mathbf{Q}_r\mathbf{v}\ . \tag{A.9}$$

Substituting Eqs. A.8 and A.9 into Eq. A.2 gives the unconstrained non-linear least squares problem:

$$\min_{\mathbf{v}} \|\mathbf{A}\mathbf{H}^T\mathbf{d} + \mathbf{A}\mathbf{Q}_r\mathbf{v} - \mathbf{c}(H^T\mathbf{d} + \mathbf{Q}_r\mathbf{v})\|_2 \tag{A.10}$$

where $\mathbf{v}$ is the only unknown and $\mathbf{c}$ is a non-linear vector function.

Eq. A.10 is solved using iterative Gauss-Newton Method, which is also adapted in [68] for solving 2D shape deformation:

$$\min_{\mathbf{v}_k} \|\mathbf{A}\mathbf{H}^T\mathbf{d} + \mathbf{A}\mathbf{Q}_r\mathbf{v}_k - \mathbf{c}(H^T\mathbf{d} + \mathbf{Q}_r\mathbf{v}_{k-1})\|_2 \tag{A.11}$$

where $\mathbf{v}_k$ is the unknown at the k-th iteration; $\mathbf{v}_{k-1}$ is the solution from the previous iteration. Since $\mathbf{v}_{k-1}$ is know at the k-th iteration, $\mathbf{v}_k$ can be obtained by solving the linear system:

$$\mathbf{v}_k = (\mathbf{M}^T\mathbf{M})^{-1}\mathbf{M}^T(\mathbf{c}(H^T\mathbf{d} + \mathbf{Q}_r\mathbf{v}_{k-1}) - \mathbf{A}\mathbf{H}^T\mathbf{d}) \tag{A.12}$$

where $\mathbf{M} = \mathbf{A}\mathbf{Q}_r$. Since $\mathbf{M}$ is a fixed sparse symmetrical matrix, $(\mathbf{M}^T\mathbf{M})^{-1}\mathbf{M}^T$ can be precomputed. Then for each iteration the computation required is only a sparse matrix-vector multiplication, which can be solved in $O(n)$ time.

Bibliography

[1] A. Al-khalifah, R. McCrindle, and V. N. Alexandrov. Immersive open surgery simulation. In *Proceedings of the International Conference on Computational Science*, pages 868–871, 2006.

[2] G. Bianchi, M. Harders, and G. Székely. Mesh topology identification for mass-spring models. In *MICCAI '03: Proceedings of the 6th International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 50–58, 2003.

[3] D. Bielser and M. Gross. Open surgery simulation. In *Proceeding of the 10th Medicine Meets Virtual Reality*, pages 57–62, 2002.

[4] H. Bourquain, A. Schenk, F. Link, B. Preim, G. Prause, and H.-O. Peitgen. Hepavision2 – a software assistant for preoperative planning in living-related liver transplantation and oncologic liver surgery. In *Proceedings of the Computer Assisted Radiology and Surgery*, pages 342–346, 2002.

[5] M. Bro-Nielsen, J. Tasto, R. Cunningham, and G. Merril. PREOP endoscopic simulator: A PC-based immersive training system for bronchoscopy. In *Proceedings of the 7th Medicine Meets Virtual Reality*, pages 76–82, 1999.

[6] M. Chabanas, C. Marecaux, Y. Payan, and F. Boutault. Models for planning and simulation in computer assisted orthognatic surgery. In *MICCAI '02: Proceedings of the 5th International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 315–322, 2002.

[7] J. R. Charpie. Transposition of the great arteries, eMedicine from WebMD, http://www.emedicine.com/ped/topic2548.htm, 2007.

[8] D. T. Chen, I. A. Kakadiaris, M. J. Miller, R. B. Loftin, and C. Patrick. Modeling for plastic and reconstructive breast surgery. In *MICCAI '00: Proceedings of the 3rd International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 1040–1050, 2000.

[9] K.-S. Choi. Interactive cutting of deformable objects using force propagation approach and digital design analogy. *Computers & Graphics*, 30(2):233–243, 2006.

[10] Cincinnati Children's Hospital Medical Center. Hypoplastic left heart syndrome norwood operation. http://www.cincinnatichildrens.org/health/heart-encyclopedia/anomalies/hlhs.htm, [Online; accessed 10-January-2008].

[11] S. Coquillart. Extended free-form deformation: A sculpturing tool for 3D geometric modeling. In *Proceedings of ACM SIGGRAPH*, pages 187–196, 1990.

[12] S. Cotin, H. Delingette, and N. Ayache. A hybrid elastic model for real-time cutting, deformations, and force feedback for surgery training and simulation. *The Visual Computer*, 16(7):437–452, 2000.

[13] J. R. Crouch, J. C. Merriam, and E. R. Crouch. Finite element model of cornea deformation. In *MICCAI '05: Proceedings of the 8th International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 591–598, 2005.

[14] M. de Berg, M. van Kreveld, M. Overmars, and O. Schwarzkopf. *Computational Geometry: Algorithms and Applications*. Springer, 2nd edition, 2000.

[15] M. Desbrun, M. Meyer, P. Schröder, and A. H. Barr. Implicit fairing of irregular meshes using diffusion and curvature flow. In *Proceedings of ACM SIGGRAPH*, pages 317–324, 1999.

[16] J. Dutreuil, F. Goulette, C. Laurgeau, J. C. Zoreda, and S. Lundgren. Computer assisted dental implantology: A new method and a clinical validation. In *MICCAI '01: Proceedings of the 4th International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 384–391, 2001.

[17] M. A. ElHelw, M. S. Atkins, M. Nicolaou, A. Chung, and G.-Z. Yang. Photo-realistic tissue reflectance modelling for minimally invasive surgical simulation. In *MICCAI '05: Proceedings of the 8th International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 868–875, 2005.

[18] L. France, J. Lenoir, A. Angelidis, P. Meseure, M.-P. Cani, F. Faure, and C. Chaillou. A layered model of a virtual human intestine for surgery simulation. *Medical Images Analysis*, 9(2):123–132, 2005.

[19] C. C. Galanis, M. M. Sfantsikopoulos, P. T. Koidis, N. M. Kafantaris, and P. G. Mpikos. Computer methods for automating preoperative dental implant planning: Implant positioning and size assignment. *Computer Methods and Programs in Biomedicine*, 86(1):30–38, 2007.

[20] T. J. Gardner and T. L. Spray. *Operative Cardiac Surgery*. Arnold, 5 edition, 2004.

[21] E. Gladilin, A. Ivanov, and V. Roginsky. Generic approach for biomechanical simulation of typical boundary value problems in cranio-maxillofacial surgery planning. In *MICCAI '04: Proceedings of the 7th International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 380–388, 2004.

[22] O. Goksel, S. E. Salcudean, and S. P. Dimaio. 3D simulation of needle-tissue interaction with application to prostate brachytherapy. *Computer Aided Surgery*, 11(6):279–288, 2006.

[23] M. Hauth, O. Etzmuss, and W. Strasser. Analysis of numerical methods for the simulation of deformable models. *The Visual Computer*, 19(7-8):581–590, 2003.

[24] A. Kimura, J. J. Camp, R. A. Robb, and B. Davis. A prostate brachytherapy training rehearsal system - simulation of deformable needle insertion. In *MICCAI '02: Proceedings of the 5th International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 264–271, 2002.

[25] Kitware, Inc. The Visualization Toolkit. `http://www.vtk.org`.

[26] U. Kuhnapfel, H. Cakmak, and H. Maass. Endoscopic surgery training using virtual reality and deformable tissue simulation. *Computers & Graphics*, 24:671–682, 2000.

[27] F. Lacour-Gayet. Anatomical repair of transposition of the great arteries. In T. J. Gardner and T. L. Spray, editors, *Operative Cardiac Surgery*. Arnold, 5 edition, 2004.

[28] L. L. Lian and Y. H. Chen. Haptic surgical simulation: An application to virtual suture. *Computer-Aided Design & Applications*, 3(1-4):203–210, 2006.

[29] A. Liu, C. Kaufmann, and D. Tanaka. An architecture for simulating needle-based surgical procedures. In *MICCAI '01: Proceedings of the 4th International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 1137–1144, 2001.

[30] A. Liu, F. Tendick, K. Cleary, and C. Kaufmann. A survey of surgical simulation: applications, technology, and education. *Presence: Teleoperators & Virtual Environments*, 12(6):599–614, 2003.

[31] R. MacCracken and K. I. Joy. Free-form deformations with lattices of arbitrary topology. In *Proceedings of ACM SIGGRAPH*, pages 181–188, 1996.

[32] H. Masuda, Y. Yoshioka, and Y. Furukawa. Interactive mesh deformation using equality-constrained least squares. *Computers & Graphics*, 30(6):936–946, 2006.

[33] H.-P. Meinzer, M. Thorn, and C. Cardenas. Computerized planning of liver surgery: an overview. *Computers & Graphics*, 26(4):569–576, 2002.

[34] M. Meyer, M. Desbrun, P. Schrder, and A. H. Barr. Discrete differential-geometry operators for triangulated 2-manifolds. In H.-C. Hege and K. Polthier, editors, *Visualization and Mathematics III*, pages 35–57. Springer-Verlag, 2002.

[35] W. Mollemans, F. Schutyser, J. V. Cleynenbreugel, and P. Suetens. Fast soft tissue deformation with tetrahedral mass spring model for maxillofacial surgery planning systems. In *MICCAI '04: Proceedings of the 7th International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 371–379, 2004.

[36] A. G. Moon. Transposition of the great arteries. *Journal of Diagnostic Medical Sonography*, 19(6):375–378, 2003.

[37] J. Mosegaard. LR-spring mass model for cardiac surgical simulation. In *Proceedings of the 12th Medicine Meets Virtual Reality*, pages 256–258, 2004.

[38] J. Mosegaard, P. Herborg, and T. S. Sørensen. A GPU accelerated spring mass system for surgical simulation. In *Proceedings of the 13th Medicine Meets Virtual Reality*, pages 342–348, 2005.

[39] J. Mosegaard and T. S. Sørensen. Real-time deformation of detailed geometry based on mappings to a less detailed physical simulation on the GPU. In *Proceedings of the 11th Eurographics Workshop on Virtual Enviroments*, pages 105–111, 2005.

[40] M. Muller, M. Teschner, and M. Gross. Physically-based simulation of objects represented by surface meshes. In *Proceedings of the Computer Graphics International*, pages 26–33, 2004.

[41] A. Nealen, M. Müller, R. Keiser, E. Boxermann, and M. Carlson. Physically based deformable models in computer graphics. In *Proceedings of the Eurographics 2005, State of the Art Reports*, pages 71–94, 2005.

[42] H.-W. Nienhuys and A. F. van der Stappen. A surgery simulation supporting cuts and finite element deformation. In *MICCAI '01: Proceedings of the 4th International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 153–160, 2001.

[43] H.-W. Nienhuys and A. F. van der Stappen. A Delaunay approach to interactive cutting in triangulated surfaces. In *Fifth International Workshop on Algorithmic Foundations of Robotics*, pages 113–129, 2003.

[44] J. D. Owens, D. Luebke, N. Govindaraju, M. Harris, J. Krger, A. E. Lefohn, and T. J. Purcell. A survey of general-purpose computation on graphics hardware. In *Proceedings of the Eurographics 2005, State of the Art Reports*, pages 21–51, 2005.

[45] M. A. Padilla Castañeda and F. Arámbula Cosío. Deformable model of the prostate for TURP surgery simulation. *Computers & Graphics*, 28(5):767–777, 2004.

[46] S. Park, X. Guo, H. Shin, and H. Qin. Surface completion for shape and appearance. *The Visual Computer*, 22:168–180, 2006.

[47] Y. Qi. Segmentation and reconstruction of 3d artery models for surgical planning. Master's thesis, School of Computing, National University of Singapore, 2008.

[48] G. Ranzuglia, P. Cignoni, F. Ganovelli, and R. Scopigno. Implementing mesh-based approaches for deformable objects on GPU. In *Proceedings of the 4th Eurographics Italian Chapter 2006*, pages 213–218, 2006.

[49] D. Richens, M. Fielda, S. Hashima, M. Nealeb, and C. Oakley. A finite element model of blunt traumatic aortic rupture. *European Journal of Cardio-Thoracic Surgery*, 25(6):1039–1047, 2004.

[50] M. Rieger, M. Gabl, H. Gruber, W. R. Jaschke1, and A. Mallouhi. CT virtual reality in the preoperative workup of malunited distal radius fractures: preliminary results. *European Radiology*, 15(4):792–797, 2004.

[51] T. W. Sederberg and S. R. Parry. Free-form deformation of solid geometric models. In *Proceedings of ACM SIGGRAPH*, pages 151–160, 1986.

[52] R. Sierra, J. Zátonyi, M. Bajka, G. Székely, and M. Harders. Hydrometra simulation for vr-based hysteroscopy training. In *MICCAI '05: Proceedings of the 8th International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 575–582, 2005.

[53] E. J. Smith, J. T. Bryant, and R. E. Ellis. Kinematic geometry of osteotomies. In *MICCAI '05: Proceedings of the 8th International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 902–909, 2005.

[54] T. S. Sørensen and J. Mosegaard. An introduction to GPU accelerated surgical simulation. In *Proceedings of the 3rd International Symposium on Computational Models for Biomedical Simulation*, pages 93–104, 2006.

[55] T. S. Sørensen and J. Mosegaard. Virtual open heart surgery: training complex surgical procedures in congenital heart disease. In *Proceeding of ACM SIGGRAPH*, page 35, 2006.

[56] T. S. Sørensen, J. Stawiaski, and J. Mosegaard. Virtual open heart surgery: obtaining models suitable for surgical simulation. In *Proceedings of the 15th Medicine Meets Virtual Reality*, pages 445–447, 2007.

[57] O. Sorkine, Y. Lipman, D. Cohen-Or, M. Alexa, C. Rssl, and H.-P. Seidel. Laplacian surface editing. In *Procedings of EurographicsACM SIGGRAPH Symposium on Geometry Processing*, pages 175–184, 2004.

[58] G. Szekely, C. Brechbuehler, R. Hutter, A. Rhomberg, and P. Schmid. Modeling of soft tissue deformation for laparoscopic surgery simulation. In *MICCAI '98: Proceedings of the 1st International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 550–561, 1998.

[59] P. Tardieu, L. Vrielinck, E. Escolano, M. Henne, and A.-L. Tardieu. Computer-assisted implant placement: scan template, SimPlant, SurgiGuide, and SAFE system. *The International Journal of Periodontics & Restorative Dentistry*, 27(2):141–149, 2007.

[60] G. Taubin. A signal processing approach to fair surface design. In *Proceedings of ACM SIGGRAPH*, pages 351–358, 1995.

[61] C. I. Tchervenkov and S. J. Korkola. Transposition of the great arteries with left-or-right ventricular outflow tract obstruction. In T. J. Gardner and T. L. Spray, editors, *Operative Cardiac Surgery*. Arnold, 5 edition, 2004.

[62] M. Thubrikar. *The Aortic Valve*. CRC Press, Inc., 1990.

[63] S. Toledo, D. Chen, and V. Rotkin. Taucs: a linrary of sparse linear solvers, 2003. `http://www.tau.ac.il/stoledo/taucs/`.

[64] C. Y. Tso, R. E. Ellis, J. Rudan, and M. M. Harrison. A surgical planning and guidance system for high tibial osteotomies. In *MICCAI '98: Proceedings of the 1st International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 39–50, 1998.

[65] M. Ursino, J. L. Tasto, B. H. Nguyen, R. Cunningham, and G. L. Merril. Cathsim: an intravascular catheterization simulator on a PC. In *Proceedings of the 7th Medicine Meets Virtual Reality*, pages 360–366, 1999.

[66] K. Verstreken, J. van Cleynenbreugel, G. Marchal, D. van Steenberghe, and P. Suetens. Computer-assisted planning of oral implant surgery - an approach using virtual reality. In *Proceedings of the 4th Medicine Meets Virtual Reality*, pages 423–434, 1996.

[67] R. W. Webster, D. I. Zimmerman, B. J. Mohler, M. G. Melkonian, and H. R. S. A prototype haptic suturing simulator. *Stud Health Technol Inform*, 81:567–569, 2001.

[68] Y. Weng, W. Xu, Y. Wu, K. Zhou, and B. Guo. 2D shape deformation using nonlinear least squares optimization. *The Visual Computer*, 22(9):653–660, 2006.

[69] C. Williams, I. A. Kakadaris, K. Ravi-Chandar, M. J. Miller, and C. W. Patrick. Simulation studies for predicting surgical outcomes in breast reconstructive surgery. In *MICCAI '03: Proceedings of the 6th International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 9–16, 2003.

[70] J. R. Wollmuth, , D. R. Bree, B. P. Cupps, M. D. Krock, B. J. Pomerantz, R. P. Pasquea, A. Howells, N. Moazami, N. T. Kouchoukos, and M. K. Pasque. Left ventricular wall stress in patients with severe aortic insufficiency with finite element analysis. *The Annals of Thoracic Surgery*, 82(3):840–846, 2006.

[71] X. Wu, M. S. Downes, T. Goktekin, and F. Tendick. Adaptive nonlinear finite elements for deformable body simulation using dynamic progressive meshes. In *Proceedings of the Eurographics 2001*, pages 349–358, 2001.

[72] J. Xia, N. Samman, C. K. Chua, R. W. K. Yeung, D. Wang, S. G. Shen, H. H.-S. Ip, and H. Tideman. PC-based virtual reality surgical simulation for orthognathic surgery. In *MICCAI '00: Proceedings of the 3th International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 1019–1028, 2000.

[73] Y. Yu, K. Zhou, D. Xu, X. Shi, H. Bao, B. Guo, and H.-Y. Shum. Mesh editing with Poisson-based gradient field manipulation. *ACM Transactions on Graphics*, 23(3):644–651, 2004.

[74] W. Zhao, S. Gao, and H. Lin. A robust hole-filling algorithm for triangular mesh. *The Visual Computer*, 23(12):437–452, 2007.