WEB-BASED MEDICAL TRAINING SYSTEM AND SURGICAL SIMULATOR FOR INTERVENTIONAL NEURORADIOLOGY PROCEDURES

LU YIPING

NATIONAL UNIVERSITY OF SINGAPORE

2004

WEB-BASED MEDICAL TRAINING SYSTEM AND SURGICAL SIMULATOR FOR INTERVENTIONAL NEURORADIOLOGY PROCEDURES

LU YIPING
(M.ENG., HUST)

A THESIS SUBMITTED

FOR THE DEGREE OF MASTER OF SCIENCE

SCHOOL OF COMPUTING

NATIONAL UNIVERSITY OF SINGAPORE

2004

Acknowledgements

I would like to thank the numerous people who have offered support, advice and encouragement during my graduate studies.

My deepest gratitude goes to my supervisor Prof. Loe KiaFock and Ma Xin for their guidance and support. They have always been patient and encouraging to my research work

I would also thank to Dr. Nowinski and Chui CheeKong. As the director of our laboratory, Dr. Nowinski had provided me an excellent environment for my graduate studies. He also gave me a lot of advices on my research work. My former supervisor, CheeKong Chui, he helped me a lot when I first began my studies and research here.

Many thanks also to my friends and colleagues who have helped me on my graduate studies: Wang Zhenlan, Zuo Wei, Gao Chunping, Li Yang, Kang Yulin and Dung Nguyen from Netherlands.

Finally, special thanks to my wife, my parents and brother whom without their love and support I could not have finished my study in NUS.

Table of Content

List of Figures

Summary

The purpose of this dissertation is to develop a web-enabled medical simulation system for interventional neuroradiology procedures. This system can provide the trainee a high fidelity virtual environment in both visualization and haptic rendering. As an excellent media for delivering information, WWW offers accessibility and distributed computing which can be used to provide novel solutions for traditional applications. Combined with Java3D, the web-based medical simulation system is independent of platform and also has scalability.

The physical-based modeling of the vascular network which is proposed in this work is an important feature. In this model, the central axis model is used to represent the human vasculature and a simplified control mesh is reconstructed for 3D visualization. The succeeding surface subdivision makes the coarse control mesh smooth and the visualization after rendering is satisfactory for the simulation system. The data structure to store the axis model is very small, which shortens the transmission time from the web server side to the training client side. The hierarchical structure of the model is also convenient to build the Oriented Bounding Box (OBB) tree for fast collision detection.

Haptic rendering in the web-based simulation system is another important feature. Although the system assumes the vasculature model and tools' model as rigid objects, it provides the trainee a relatively realistic force feedback environment over the Internet. The compact data structure and the fast collision detection algorithm make the haptic rendering a real-time one. The distributed haptic force feedback computation model in our system also let the client side focus on the quality and performance of 3D model visualization.

Chapter 1

Introduction

1.1 Research Background

Since the early 1990's, extensive attention has been paid to training medical staff in minimally invasive surgery (MIS)[1]. MIS procedures, such as interventional radiological procedures, always involve inserting a guidewire and a catheter into blood vessels and unblocking the artery to restore the blood flow. For performance of MIS procedures requires great skill to avoid complications that may cause serious injury to a patient, it is necessary for the medical staff to practice more and gain experience to make such procedure successful. However, training and education of this type of surgery is an expensive and time-consuming process, it is also a matter of close supervision on apprenticeship model [2]: In practice the trainees need to learn how to perform surgical procedures by watching experts doing. Hands on practice, using animals or models rather than patients, is minimal and unrealistic. Indeed, the real practice occurs on patients. However, the chance of doing such a practice is not very high; consequently, newly trained staffs are more susceptible to complications during a procedure, which forms an added risk. Furthermore, to evaluate the trainee, paper exams, subjective evaluation by consultants or video monitoring of procedures are used. Hence, the current situation of surgical training is as follows: there is spending of health care budget and consultant time for training without any systematic means of validating that trainee has acquired the needed skills until they perform on patients.

To solve above claimed situation, a number of solutions have been proposed. They either focus on a specific MIS application or provide a generic solution –e.g. [3,4]. Utilizing Virtual Reality (VR) technology is a shared feature of these solutions. The reason for this technology is straightforward; VR offers a realistic environment and natural methods of interaction. The related research targeting this problem has been active since 1994. However, very few results from this research has been used clinically despite huge efforts have been made to apply VR to MIS. One major reason for this is the sensitive nature of the medical field as a result of dealing with human subjects rather than objects. High fidelity is a requirement of any suggested substitute. However, virtual reality has not shown enough evidence of this aspect so far. Another reason is the complexity of the human body, which is reflected by the complicated inter-relationship between human body's components and its anatomy, physiology and psychology; this makes it difficult to represent its diverse functionalities and shapes. The complexity of the medical field added to the complexity of virtual reality interactions also push efforts in this area to high cost solutions. All these factors make virtual reality solutions limited in research laboratories.

At the same time, the World Wide Web (WWW) got rapid development and began to take a more formal shape when the World Wide Web Consortium (W3C) was founded in 1994. After that, the WWW has become a virtual society that hosts a huge amount of information. With the exponential growth in the number of web societies, it is redirecting its attention to provide services in addition to information. These services can be accessed by users with different needs, from highly specialized complex applications serving experts to applications of general utility such as e-commerce all over the world. In order to make the traditional application accessible to more people

and less expensive, many web-based technologies have been used to migrate legacy application systems to web-based application systems. Naturally, the medical field would be one excellent target for the web society. Actually, we can see the attraction between these two fields yielding some useful results already. Medical informatics has realized the potential of the web as a media, which has resulted in a number of medical web sites that concentrate on either patient-specific issues, general health interests, or medical practitioners' interests. Nevertheless, the potential of the web in the medical field has not yet been fully exploited.

1.2 Problem and objectives

The Biomedical Imaging Lab of KRDL (Kent Ridge Digital Lab) has been engaged in the research pertaining to medical simulation and educational applications [5, 6] for a long time. They have made some progresses in the field of simulation of Interventional Neuroradiology Procedures. In their real-time interactive simulator system for vascular catheterization [6, 7] project, various image-guided procedures are simulated including vascular catheterization, angioplasty, and stent placement in a realistic and interactive manner.

Based on the previous research work done by the former, we propose research and development of simulation technologies that enable realistic virtual reality training on neuroradiology procedures delivered on the web [8]. The ubiquity, easy accessibility, platform independency and low cost of web technology cooperating with the modern virtual reality technology will provide a whole new world to medical simulation system. Without gathering in a specific training room, the trainee can do interventional

neuroradiology procedure – navigating catheters, inflating balloon and deploying coils on his own office or even at home. Using our extensive experience with haptic feedback, we can also deliver force feedback to the trainee via the Internet.

Because of the ubiquitous nature of the web, the solutions provided on its environment are expected to be accessible via low-technology equipment as well as high ones. Thus, there is a trade off between accessibility (provided by the web) and realism (provided by VR). To create an acceptable solution for the trainee, other problems have to be solved firstly.

The first problem is the data size of the vasculature system and the model of tools. Because our simulation system is a web-based application, the modeling data has to be transferred from server side to the client side. To make the transmission time as short as possible and lessen the computation workload over the manipulation of the model, we should keep the vascular model and tool model data smaller compared with those of the traditional medical simulation system.

The second issue is to build a realistic and accurate visualization of the vasculature system and tools from the model. This work is essential for a creditable simulation system for interventional radiology. A small model of vasculature and tools, which do not contain all detailed surface information are used in our simulation system. A challenging question rises about how we can provide the trainee a realistic visual simulation environment under these conditions.

Real time motion tracking of tools and realistic force feedback from the tools will be the third problem we have to deal with in our simulation system. Besides the realistic visualization of medical simulation system, we also need to provide the trainee a haptic rendering that can improve the perception and understanding to augment visual simulation environment. The haptic rendering will allow trainee to reach into the virtual simulation world with a sense of touch so that they can feel and manipulate simulated objects.

This thesis presents the author's work on solving the three problems mentioned above. In the research work of image segmentation, vascular information including geometry and topology is extracted from the medical data. After the process of image segmentation, a Central Axis Model is proposed to represent human vascular networks in a compact way. The model is stored in HCLM format file (see the appendix for the detailed format of HCLM file) on the physical media. Based on the geometry and topological information provided by the HCLM file, we construct the initial control mesh and then use some certain kinds of smoothing methods to make the surface of vasculature smooth and be similar to the real human vasculature. To compute collision between the vasculature system and tools efficiently and quickly, we use tight-fitting oriented bounding boxes (OBBs) to build a bounding volume hierarchy tree to fulfill the fast and accurate collision detection for haptic interaction between the blood vessel and tools (catheter and guidewire). To achieve a more realistic behavior of the simulation procedure, physical-based modeling is also used to make the motion of tools and force feedback natural.

1.3 Contribution of thesis

There are three criteria to evaluate a medical simulator: (1) they must be realistic; (2) they must be affordable; (3) they must be validated. Only those simulators which satisfy all of these criteria will become an acceptable training curriculum for the trainees and provide an objective measure of procedural skill. In this thesis, the author's research work will be mainly focused on the aspect of being realistic and affordable. Besides these three criteria, the research work also involves partly into making a real-time visualization and haptic rendering for the trainee.

The simulation system described in this thesis provides the realistic environment from visualization and haptic force feedback. Based on the segmentation result of medical image, the thesis introduces the reconstruction of the human vasculature system from real medical data. Several methods are applied to correct errors which existed in the source and procedure of segmentation. From the point of view of visualization, the surface subdivision method is a useful method to create a smooth blood vessel. Physical modeling of the motion of the tools and collision response from the contact of the tools and blood vessel wall also adds to a realistic and natural sense of the trainee as well. Both these methods help to provide a high fidelity virtual environment for medical simulation procedure.

The criteria of affordability is met by the web-based medical simulation system by using web technology and Java3D. Java3D is an open technology developed by Sun to effectively render 3D models and provide interaction within web browsers. Java3D also enables the communication between the hardware that provides force feedback to

the trainee and simulation system. In the architecture of the web-based medical simulation system, the client side only needs a Java3D enabled web browser and a haptic box that provides force feedback. The training data and application are located at server side. This architecture provides a cost effective and affordable solution for the trainee.

Our web-based simulation system supports a real-time visualization and haptic rendering for the trainee. After training data and a control Java applet are being transferred to the client side, the visualization and tracking of the tools will make use of the computation power of the client machine and achieve a rather real-time simulation system for the trainee.

The distributed and extensible architecture is another contribution of the author. The collision detection server is designed as a COBBA server and can be distributed over the same machine or different machines. This design can provide extensible modules for a web-based system and also distribute the computation workload over different machines. To some extent this design lowers the cost of purchasing a high performance web server machine.

1.4 Thesis organization

This thesis is organized as follows:

Chapter 1 gives an introduction to the author's research work.

Chapter 2 introduces the virtual reality technology and its role in the medical field. Some web-based applications which are related to our work are also reviewed in this chapter. Although these applications offer different solutions than we do for the same problem, they have contributed to concepts and implementation of our work. A discussion is presented at the end of the chapter to address the problems in these solutions.

Chapter 3 describes the visual modeling of the human vasculature system and the tools respectively. Firstly, the problem, related work, data source of the vasculature and the tools are presented. Secondly, the 3D representation is discussed and surface subdivision method is used to make the representation smooth and realistic from the point of view of visualization. Problems encountered in our solution are also addressed in discussion section.

Chapter 4 deals with the haptic modeling of the medical simulation system. First the author introduces the haptic feedback involved in the procedure of the simulation. Subsequently the haptic rendering are discussed from two aspects: collision detection and response and the tools' behaviour. Finally is the discussion about some problems encountered in our solution.

Chapter 5 introduces the implementation of the whole system: from the system architecture design, system environment and user interface to every detailed module.

Chapter 6 describes the conclusions of our work and suggestions for future improvements.

Chapter 2

Web-based medical simulation

2.1 Virtual reality in medicine

Virtual Reality (VR) offers great potential as a technology for application systems in medicine. As recently noted by Satava and Jones [9], the advantages of virtual environments (VEs) to health care can be summarized in a single word: revolutionary. As a technology, a communication interface and an experience [10], research of VR in medicine is moving fast. From the analysis of Riva G. [11], since Jaron Lamier used VR for the first time in 1986, the need of medical staff stimulates the use of VR to visualize complex medical data, particularly during surgery and for surgery planning [12]. The fields of surgery-related application of VR can be mainly divided into three catalogues [13]: surgery training, surgery planning and computer assisted surgery (CAS). In the following sections, we will brief the current state of research in these fields.

2.1.1 Surgery training and simulation

For surgeons, there is no alternative to hands-on practice in training. The science of virtual reality provides an entirely new opportunity in the area of simulation of surgical skills using computer for training, evaluation, and eventually certification [14]. Thus a number of medical simulators have been developed to train surgeons on new and existing surgical procedures in the early 1990s [15,16,17]. These applications are

mainly based on two approaches. One approach is to develop a general solution that can be used to build different medical simulators and another approach is to provide application-specific simulators that can be used to train on certain procedures [18].

KISMET [18,19] is a package developed by Kernforschungszentrum Karlsruhe that allows the trainee to interact with the models using shutter glasses and sensored surgical instruments. In this package, deformation of tissues and remote manipulation of instruments over a LAN are also supported.

Ixion [20] is another general toolkit that provides various patient-specific data with different pathology. It can build 3D anatomical models and add deformation and collision detection.

Teleos is another authoring toolkit to assist the modeling surgical simulators. In this toolkit, it allows the representation of data to be deformation spline-based models, volume, polygonal mesh or physically-based tubes. Collision detection and organ deformation are supported and blood flow is simulated to simulate pathology. The simulator for laparoscopy can model the inflation of the catheter balloon and contrast agent injection. It can also provide different options to treat the patient [3,21,22].

Simulators that are used to train physicians on a specific procedure are also rapidly developed. These simulators have shown improved training efficiency over traditional methods. For instance, NeuroCath (see Fig 2.1) provides an efficient, comprehensive approach for clinicians to understand the complexity of the interaction of vascular structures, therapeutic devices and required techniques to perform interventional

procedures [23,24]. Another example is the Minimally Invasive Surgery Training Virtual Reality (MIST-VR) trainer [25,26], it showed that VR simulation was effective in training the novice to perform basic laparscopic skills.

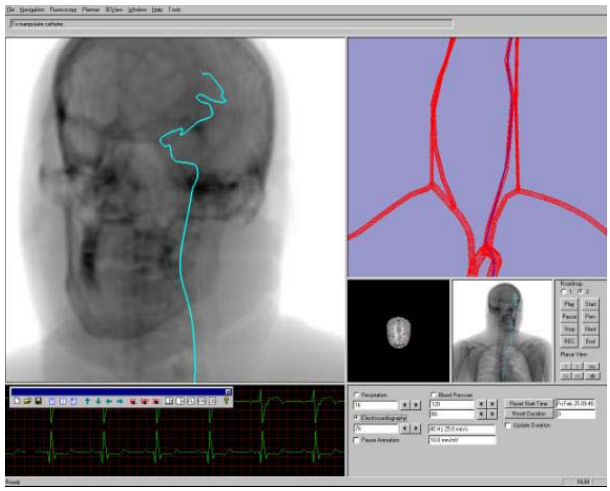


Figure 2.1 Interventional neuroradiology simulator: NeuroCath

2.1.2 Surgery planning

Surgical planning is another typical use of VR in the field of medicine. The planning of surgical and neuro-surgical procedures usually relies on the studies of two-dimensional MR (Magnetic Resonance) and/or CT (Computer Tomography). The VR-based system can incorporate different scanning modalities and provide a simple to use interactive 3D view. Cyberscalpel is a typical surgical system for planning and practice developed by NASA [27]. It can be used to plan the operation of a patient with a cancer of the jaw. In [28], a system utilizes virtual reality technology to investigate alternative operations for hip and knee orthopaedic surgery. With the fusion between CT and NMR (Nuclear Magnetic Resonance) data and using a spaceball, the user can perform measurements in traditional two-dimensional way or in three-dimensions.

2.1.3 Computer assisted surgery

Virtual reality can be utilised to provide intra-operative assistance to the medical staff. Taking the form of fusion of intra-operative data, most intra-operative assistance can facilitate the accomplishment of the target. An image-guided surgery system presented in [29] utilizes HMD (head mounted display) and magnetically sensored tools to correlate between intra-operative MR scans and pre-operative CT scans as well as showing the position of the tools.

2.2 Web-based virtual reality in surgical training

In [30], the author found that the World Wide Web could provide a suitable environment for surgical simulators, particularly for many of the procedural specialities carried out by surgeons, endoscopists, interventional radiologists, and interventional cardiologists. Using the WWW as the delivery mechanism guarantees that the requirement of affordable simulators can be met. Utilizing the Virtual Reality Modelling Language (VRML) and Java can also enable render the 3D anatomical models and interaction over the WWW effectively. Till now only a few research groups are investigating the use of the WWW for surgical training and planning. In Manchester Visualization Center, the researchers have implemented the Web-based Standard Educational Tools (WebSTer) [1] environment with VRML and Java and use it to develop many applications for many medical procedures training, such as simulator for treating abdominal aortic aneurysms [18], ventricular catheterisation [31] and lumbar puncture [32]. The ASPIRE (Advanced Surgical Planning – Interactive Research Environment) project [33, 34] is also using a 3D web-based front end, for

pre-operative surgical planning of vascular reconstruction. Figure 2.2 is a web-based lumbar puncture simulator that is involved in inserting a needle between vertebrae in the lower back directly into the spinal cord and taking a sample of the spinal fluid for various tests.

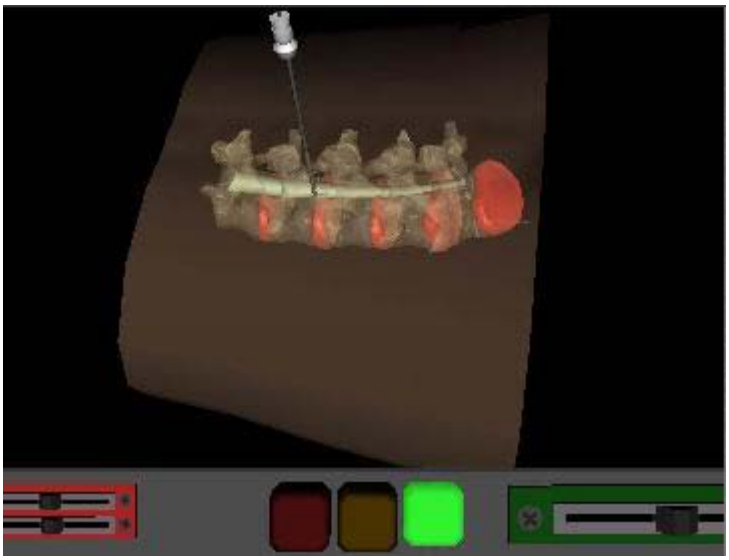


Figure 2.2. Web-based Lumbar Puncture Simulator

2.3 Discussion

Most web-based realistic virtual environment systems are built with VRML and Java. These implementations satisfy the realism criterion and also meet the given constraints for web-based delivery. Compared with those expensive high fidelity surgical simulators, these solutions provide cost-effective and easy to access VR-based medical simulation environment and will introduce many new surgeons and clinicians to the potential of VR in medicine. The implemented prototypes show that WWW can provide an effective virtual environment within which training can be enhanced by 3D simulation and interaction [35].

However, current web-based VR technology suffers from some limitations:

1. Accuracy. With regarding to the model ability of VRML, it is not easy to represent the complicated human organs with VRML. High fidelity human organ model often contains bigger data information and it is not practicable to render the accurate 3D model in VRML-enabled web browser. In the virtual training environment, accuracy will be compromised by the more important realism.
2. Interaction. Current VRML-based solution needs a VRML plug-in for the web browser. With the limited supporting for Java to maneuver the VRML model, this will limit the interaction between the trainee and the simulation system.
3. Haptic feedback. In most web-based simulation systems, a mouse or keyboard are the only input devices. Although some systems provide force feedback mouse and support haptic rendering, the realism of haptic is simple and can't simulate the force feedback in real surgical procedure.

Despite the limitations in these solutions, the web technology provides the virtual simulation system a scalable and portable architecture. The research of our work will follow the basic idea of these solutions and focus on resolving the major question of degree of realism.

Chapter 3

Visual modeling

From the vessel segmentation result on medical image, we can obtain the skeletal information of the vascular networks. To render and manipulate the vessel in our simulation system, we need to model and reconstruct the vascular system in three dimensions from the segmentation result. Besides the human vascular system, we also need to model and reconstruct the tools. This chapter describes the visual modeling of the vascular system and the tools. Section 3.1 gives an introduction of research in visualization of vasculature. Section 3.2 describes the details of visual modeling of vasculature in our system. Section 3.3 briefs the modeling method of the tools. Problems are discussed in section 3.4

3.1 Introduction

The computer-assisted visualization of anatomical structures reconstruction from a set of 2D images such as Computer Tomographies (CT) and Magnetic Resonances (MR) is becoming an increasingly relevant feature of the medical diagnosis and the medical simulations [36]. The process of 3D vasculature visualization often consists of the following stages:

1. Segment the vasculature structure from medical images.
2. Model the topological and geometrical information and reconstruct it.
3. Render the model and visualize it.

The process of segmentation is to identify blood vessel regions in medical image and group them together. The output of segmentation is usually a set of classified elements, such as the contour of the blood vessels. Numerous approaches that are suitable for identifying blood vessels have been proposed. According to [37,38], these methods can be classified into three categories: manual, semi-automatic and fully automatic. Although manual segmentation is extremely time-consuming and needs the knowledge of anatomy, its result owns high quality. To save the labor of manual segmentation, semi-automatic and fully automatic segmentation algorithms are also developed to detect the narrow tubular structures with low contrast and small objects causing signal attenuation [39,40,41,42]. With the interference of use-interaction, the segmentation result can be enhanced in the case of signal noise, drift in image intensity and lack of image contrast. In those automatic algorithms, region-growing [43,44,45] method is used widely. This algorithm selects a seed point and grows several regions. Those regions that are connected and show sufficiently similarity are merged. Combined with local-threshold method, this method can be a robust and convincing approach.

After the segmentation, skeletonization methods based on topological thinning processes [46,47,48], distance transformation [49,50], or Voronoi diagrams [51,52] are applied. These methods can extract the vessel tree information, preserving the topology of the original shape and approximate the central axis. They are thin, smooth and continuous [53].

How to model the vessel structure in 3D will be the successive problem after the extraction process? There are mainly two methods to model the vessel structure: 1) Physical model. 2) Mathematical model.

3.1.1 Physical Model

The human vascular system is a tree of blood vessels with multiple branchings. In general, the system can be considered as a connected structure although there exist some isolated vascular structures in some malformations.

The physical model characterizes the vascular system and defines the properties to be analyzed: Depending on the shape of the vessels cross-sections and on the variation of the area along the main direction of the blood flux, two types of vessels may be distinguished: regular vessels and abnormal vessels. It is supposed that the cross-section of the vessels in which the area is measured and the shape of the section is evaluated, are perpendicular to the main direction of the flux. According to the different shape of the structure, the blood vessel can be classified into following category [36]:

1. Regular segment: segment of a vessel such that it is a tubular narrow structure, connected, smooth and with an approximately circular cross-section [54]

2. Stenosis or constriction: segment in which a local minimum exists in the area differential along the vascular axis. This definition encloses both asymmetrical and symmetrical stenosis.

3. Aneurysm or dilation: segment in which a local maximum exists in the area differential along the vascular axis. Similarly to the stenoses, aneurysms may be symmetrical or not with regard to the main flux direction.

3.1.2 Mathematical Model

The physical structure of the vasculature is tree-like. Therefore, an abstract tree with nodes representing junctions and edges representing tubes presents itself as the obvious choice for the logical definition of the vascular area. In the mathematical model, the structure of the vasculature is naturally represented by a graph (*n*, *e*), where *n* are the nodes of the graph which represent the branchings of the vascular map and *e* is the edges between branchings, composed of a set of segments either normal or abnormal.

Blood vessels may be considered as closed objects resulting from a cylindrical sweep in which there are no intersections neither in the contours nor in the profiles. Therefore in [55], the author used the generalized cylinders to define the mathematical model of the vascular map surface. The generalized cylinders represent each segment of the vascular tree.

3.2 Vasculature modeling

Among all the components of creating a virtual reality training environment, modeling plays the most important role: whether it is good or not is related to the quality of the whole application system. Furthermore, the medical simulation system often deals with a wide range of organs. These different organs preserve various geometric and behaviour properties. How to model components in the system and reuse them are becoming an emerging problem in simulation field. In this chapter, only the visual model is discussed, which means how to visualize the vasculature model. In our system, both physical model and mathematical model are used and the procedure is

divided into three stages according to Barzel's method [56]: First is the creation of an abstract or conceptual model of the object, where a list of the properties of the object is prepared. Second, the abstraction is described in a formal and concise way. Finally, the model is implemented in some media. In our system, we will follow these three stages to build the vascular model. Figure 3.1 shows the three stages in visual modeling vascualture in our system.

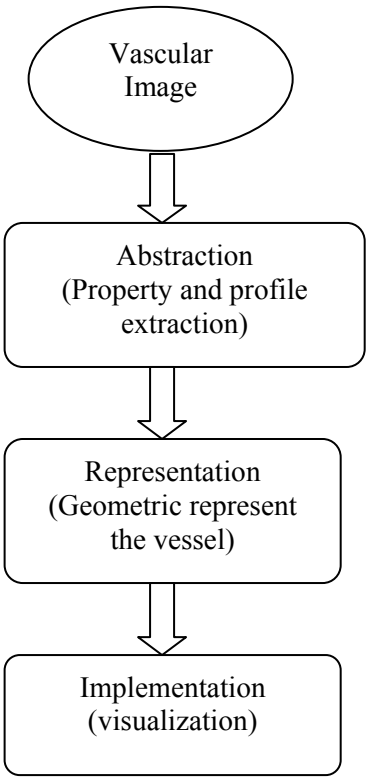


Figure 3.1 Three stages in model the vasculature

3.2.1 Abstraction of vascular system

In our system, the abstraction stage of vasculature consists of extraction of the geometrical properties and physical properties. Blood vessels are thin-walled tubes with branching structures. Its geometrical properties include 1) Diameter; 2) Thickness; 3) Branching structures. For blood vessels are composed of soft tissues, the vessel wall

will deform under the pressure of the blood flow as well as the tools insert into the vessel and collide with the blood vessel wall. It is not easy to extract these properties and to model them, because the physical elasticity property is non-linear and it varies on different parts of the body. We will only discuss the extraction process of the geometrical properties.

This part of work is done by the author's fellow research members in Biomedical Imaging Labs [57]. The purpose of this part of work is to use the semi-automatic segmentation method to extract the initial skeletal information of the vasculature. The following briefs the basic idea of the work.

The medical images, such as CT and MRI images, contain structures of the human body. The image segmentation method used in the project differs from other segmentation methods. This method segments vasculature in a three dimensional environment rather than in each 2D slice. Its advantage is having the depth information and overall shape with the whole vascular network visualized. Figure 3.2 shows the processing pipeline of this method.

Building a virtual environment for visualization and interaction vasculature in volume data obtained through MRA or CTA (Computed Axial Tomography) is the first step. In the process, a six-degree of freedom (DOF) reach-in device is used as manipulating tool to interact with the virtual environment.

The next step is to select the dominant cross sections along the vascular network for segmentation. By controlling the reach-in device, we can modify the selection. A

"detector" with a handle and an embedded facet is rendered in the virtual environment to represent the reach-in device. After the user selects a dominant cross section, the vessel passes through the detector's facet and is almost perpendicular to it. Meanwhile the cross section is rendered in another viewpoint.

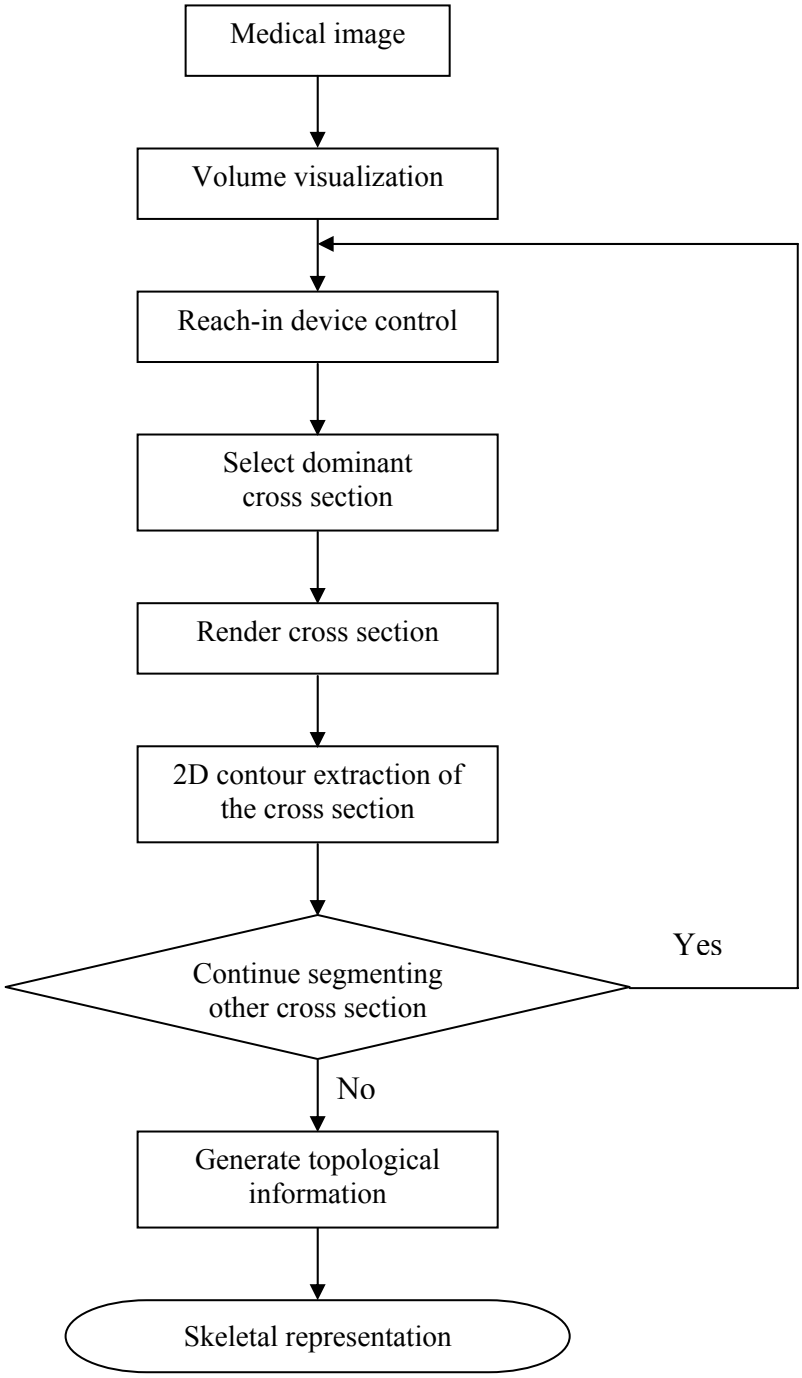


Figure 3.2 Flow chart of the semi-automatic segmentation method

2D image segmentation process is applied when a cross section is selected. The contour of that cross section will be extracted accordingly. Two methods of 2D extraction have been adopted here: the first is to use an ellipse to fit the cross section based on normal image processing technique. The other method uses a deformable model to extract a polygon description. The first method assumes that the vessel is elliptic, although it is not exactly accurate it's simple to be implemented.

After sufficient cross sections are segmented, they can form a skeleton of the vascular network. An interpolating method is also applied to reduce the user's interaction: For a certain part of vascular network, the user only need to select the beginning and ending cross section, the intermediate cross section are selected automatically. A moving facet steps forward to the next point from the beginning one, then resamples the volume data information to generate one cross section. Then the corresponding contour is extracted from the cross section image and added to the initial model. Based on the center of the extracted contour, the path to the next point is computed. When enough cross sections have been segmented, the procedure terminates. Figure 3.3 is part of the segmented cross section.

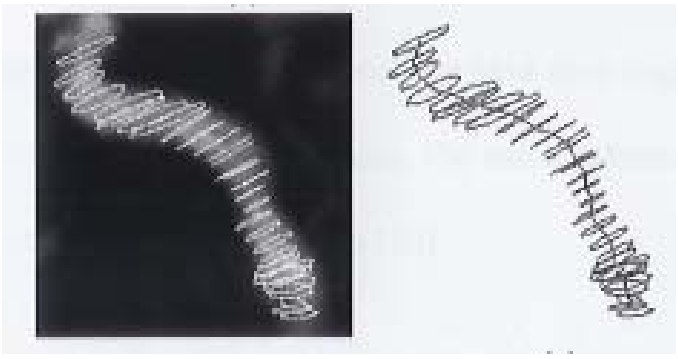


Figure 3.3 Segmented cross section

3.2.2 Representation of vascular system

After the abstraction procedure, the necessary geometric and topological information have been extracted from the medical images. Since the vasculature is a tree-like structure, we need an effective and compact representation to organize and store the segmented vasculature information. The quality of the representation is related to the subsequent model reconstruction procedure and haptic modeling will be discussed in Chapter 4. In our work, a central axis model is used to represent the result of segmentation.

The central axis model is defined as a sequence of line segments ΣS , where S consists of a sequence of nodes. A central point and a cross section around the point represent each node. The definition is as follows:

Model = $\{ \Sigma S \}$, S denotes a line segment

$S = \{ \Sigma N \}$, where N denotes a node

$N = <P, CS>$, where P is the central point and CS is the cross section

Because of the tree-like structure of vascular system, the central axis model also need to represent the topological connectivity relationship between each segment as well as the geometry information, fluid and pathology. In our central axis model, a hierarchical representation is proposed to organize the vascular system in the following structure of topology, geometry, mechanics and pathology [58]. The described topology uses a tree structure to represent the parent-child relationship between the vascular segments. Other information, such as the skeletal curves, node coordinates, fluid flow, texture and material properties are described in the level of vascular geometry and mechanics.

The topology of the vascular network is described with a hierarchical structure; it looks similar to a tree data structure. Figure 3.4 shows the topological representation of sample vasculature (left part).

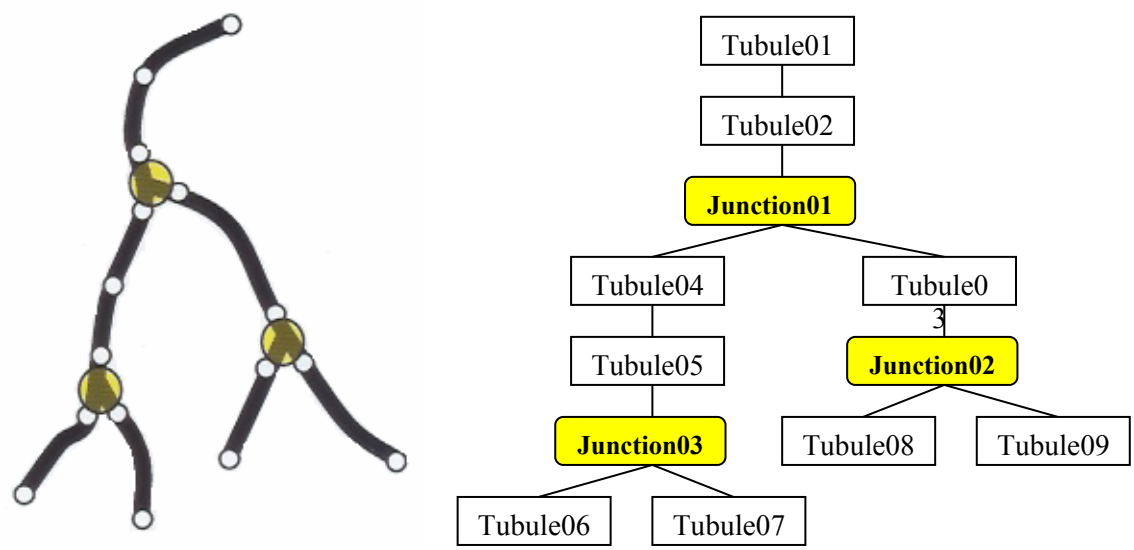


Figure 3.4 Hierarchical structures representing the topology of the vasculature

The central axis model describes the vascular networks by segments. The segment is the basic element of the vascular structure, where each vascular segment owns a unique ID. Although it is more realistic to assume the shape of the cross section of human blood vessel to be elliptic, the model is simplified by representing the cross section with a centerline point and a radius. Figure 3.5 is the internal data structure of our central axis model.

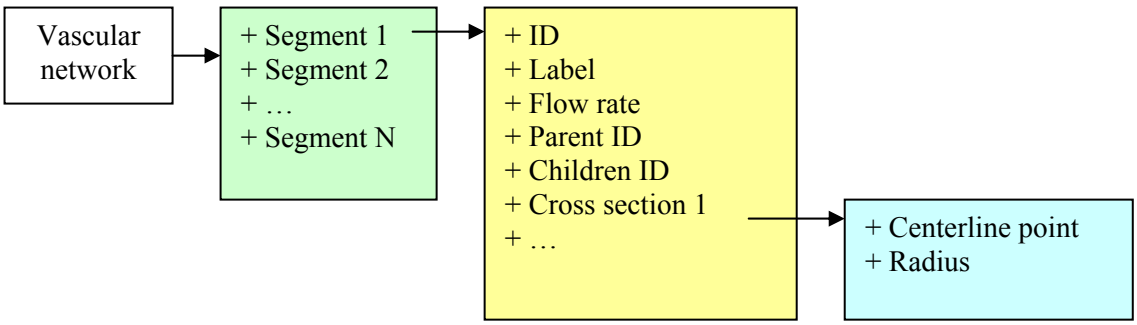


Figure 3.5 Internal data structure of the model

The physical representation of the central axis model is a HCLM format file. The detailed information can be found at Appendix A. The vessel showed in Figure 3.6 and Figure 3.7 is selected from the datasets of Biomedical Imaging Labs NeuroCath [23] project, and is the blood vessel from the human brain. Figure 3.6 shows the central line of the blood vessel and parent-child relationship within each vascular segment, which is marked with its unique ID. Figure 3.7 is the cross sections view of Figure 3.6.

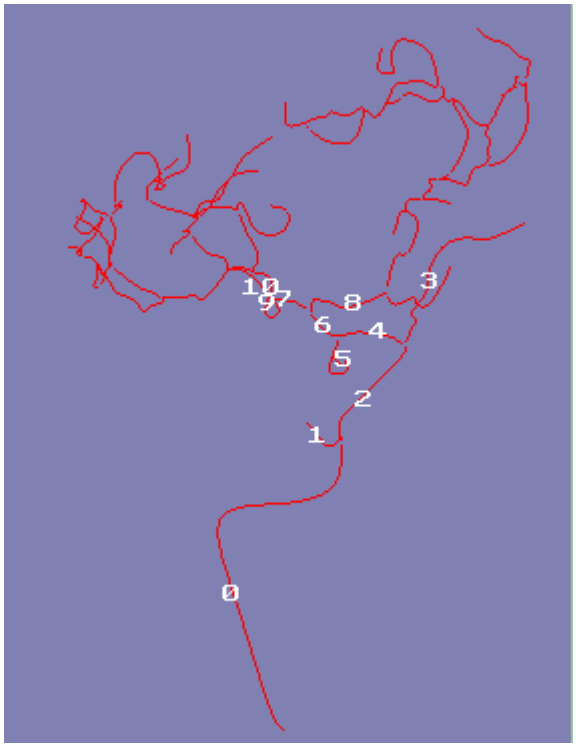


Figure 3.6 Skeletal view of the vessel

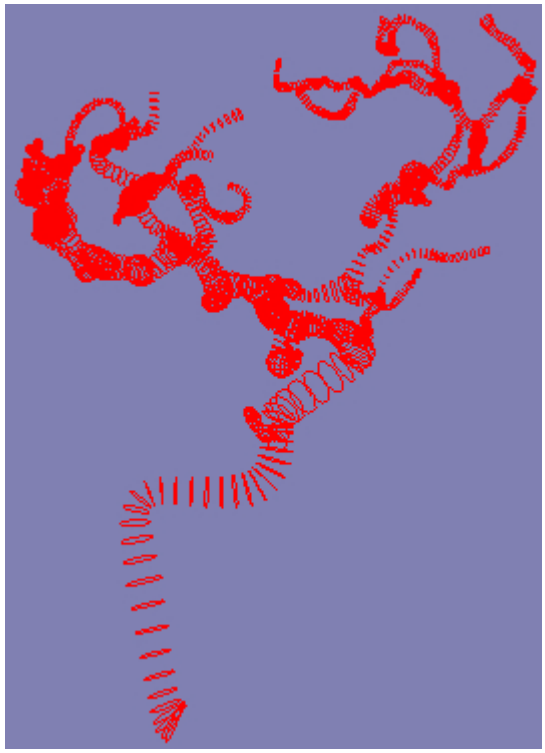


Figure 3.7 Cross section view of vessel

The central axis model provides a compact representation of the topological and geometrical information of the vascular network. It is the input of our visual vascular model and the start of reconstruction and visualization which will be discussed in the next section.

3.2.3 Model reconstruction

The central axis model contains topological and geometrical information of the vascular networks. Although it is extracted from the real human medical image, it is necessary to reconstruct the original vessel data from the central axis model and provide interaction with other media. As described in the previous section, the central axis model is a simplified representation of real human vascular network, so the reconstructed vascular network will be also a simplified representation of the real human vascular network. For a virtual reality simulation system, it is more important to be more realistic than accurate. The simplified model is not only adequate to visualize the human vascular network but also compact to deliver high performance graphic rendering over the Internet. There are two methods to reconstruct the vascular model into a three-dimensional mesh model. One is surface reconstruction technique and the other is volume reconstruction technique. With regards to the characteristic of web-based application, it is more proper to reconstruct with surface mesh model. The surface representation does not only reduce the time to render the vascular network in real-time but is also easily applied to detect the collision between the blood vessel and the catheter/guidewire over the Internet.

In more detail, the reconstruction procedure can be described as follows: the surface is constructed by the skeletal line and cross section contour generation, the next step is the initial control mesh generation and finally the surface subdivision is carried out.

3.2.3.1 Construction idea

The central axis model contains the geometric information of the vascular system and can be used to generate a surface mesh. One commonly used method is to get the

vascular contour from the centerline point and the cross section at that point; then generate the surface mesh from the generated contours. This method is very easy to understand but is very challenging to reconstruct the branching junction part. One major problem is that there will be holes left in the branching mesh [61].

In our mesh generation method, we think that the vascular network consist of a series of cylinders. These cylinders are the simplified representation of the central axis model. Strictly speaking, these cylinders are not a round tube; it is a prism without the top and bottom. These cylinders are connected from one's bottom to another's top. The branching junction part consists of a special cylinder with three prisms connected in a certain form and there will be no holes left (this detail is described in section 3.2.3.2). As soon all the cylinders are generated, the surface mesh is also generated.

The initial generated mesh is composed of prisms and the surfaces formed are not smooth. We call this the control mesh. Although it is rough but it keeps the vascular network's skeletal shape. To provide a smooth and realistic three-dimensional visual vascular model for the trainee, a surface subdivision scheme is applied to the initial mesh. After subdivision and rendering, a smooth and realistic vascular network is visualized to the trainee.

The basic idea for reconstructing the vascular network is as follows:

1. Generate the control mesh from the central axis model, it is simple but keeps the shape of the vasculature.
2. Apply surface subdivision to the control mesh to obtain a smooth surface.
3. Render the subdivided mesh and provide a smooth 3D vascular network.

3.2.3.2 Control mesh generation

The control mesh is generated from HCLM file, i.e. our central axis model. In the HCLM file, only centerline point and radius of the cross section at that point are provided. To generate the control mesh, we need to compute the points around each cross section and connect those points according to a certain rule. There are three steps to get the control mesh from our central axis model:

1. Generate the trajectory of the centerline point.
2. Compute the points around the cross section at each centerline point.
3. Connect points at each cross section according to connection rules.

1. Skeletal trajectory generation

The centerline points are obtained from the image segmentation described in Section 3.2.1. Connecting the centerline point and a trajectory of the blood vessel's skeleton is formed. As the distance between each image slice is very small, it is direct to connect the successive centerline points and generate the skeletal trajectory of the vascular network.

The directly connected skeletal trajectory is somewhat scattered and not smooth from an anatomical point of view. There two reasons to explain this problem: One is because there exist noises in the segmented medical images and it makes the centerline point shifted. The other reason is that we have defined centerline point only aligned on a regular grid within voxels with integer coordinates (every voxel has only one

intensity value, and usually an integer coordinates of this voxel are the centerline point coordinates). Although the next surface subdivision scheme can make the directly connected skeletal trajectory smooth, we still need to correct some shifted centerline points and make the skeletal trajectory intrinsically smooth. The idea we used to move the shift originates from [59]. We will use a moving average filter to move the points of the centerline a bit in order to receive a smooth curve. The computational cost of the filter is linear to the number of the centerline points. Below is the description of the moving average filter.

The smoothing is achieved by replacing each voxel $u_{p,l}$ by a point $v_{p,l}$ produced by a weighted average with its neighbors. An essential necessity is to guarantee that each new point $v_{p,l}$ be located inside the original voxel (see (1)). Thus the original geometrical information contained in the segmented data is preserved.

$$v_{p,l}: \quad \left|v_{p,l}-u_{p,l}\right|_{\infty}<0.5, \qquad v_{p,l}\in Q^3, \\ \forall p\in\{1,...,P\}, \forall l\in\{\{1,...,L(1)\},...,\{1,...,L(P)\}\} \tag{1}$$

Where p is the index of the vascular line segment (totally P line segments in the vascular network), l is the index of the centerline point in the line segment (there are $L(p)$ centerline points for the p^{th} line segment).

While many filters are possible, we used a simple moving average filter with the mask (1, 2, 1). The new centerline point $v_{p,l}$ is generated from the old centerline point $u_{p,l}$ according to equation (2). It has been found to be quite effective. The index p is omitted since the procedure is repeated for each vascular line segment.

$$
\begin{aligned}
v_1 &= u_1 \\
v_2 &= \frac{1}{4}(v_1 + 2u_2 + u_3) \\
&\ldots \\
v_l &= \frac{1}{4}(v_{l-1} + 2u_l + u_{l+1}), \qquad l = 2,\ldots, L(p)-1. \qquad (2)
\end{aligned}
$$

The maximal shift may be conveniently evaluated using the infinite norm and the triangular inequality. This mask is nearly optimal in the sense that the maximal amount of shift along each of the x, y and z directions is bounded by 0.5, thus preventing the resulting smoothed skeletal trajectory from escaping voxel boundaries in any of the three directions. All calculations are done below:

$$
\begin{aligned}
\|v_l - u_l\| \le{}& \frac{1}{4^{l-1}}\|u_1 - u_l\|_\infty + \frac{2}{4^{l-1}}\|u_2 - u_l\|_\infty + \frac{9}{4^{l-1}}\|u_3 - u_l\|_\infty + \frac{9}{4^{l-2}}\|u_4 - u_l\|_\infty \\
&+\ldots + \frac{9}{4^3}\|u_{l-1} - u_l\|_\infty + \frac{9}{4^2}\|u_l - u_l\|\infty + \frac{1}{4}\|u_{l+1} - u_l\|_\infty \\
&\le \frac{1}{4} + \frac{9}{4^2}\sum_{k=1}^{l-1}\frac{k}{4^k} - \frac{l-2}{4^l} - \frac{l-1}{4^{l+1}} \\
&\le 0.5 \quad \forall l > 1. \qquad (3)
\end{aligned}
$$

In inequation (3), || || denotes the norm. In our program, we use the same moving average filter with the mask (1, 2, 1) and process described in (2). We fix first and last centerline points of the line segment and use average filter several (in program it is 3) times. For first smoothening error <= 0.5, second <= 1.5, third ~ 3.25 pixels. Figure 3.8 is an illustration of the effect of smoothing on a two dimensional example (The blue point is the original centerline point and the red one is the result after filtering.

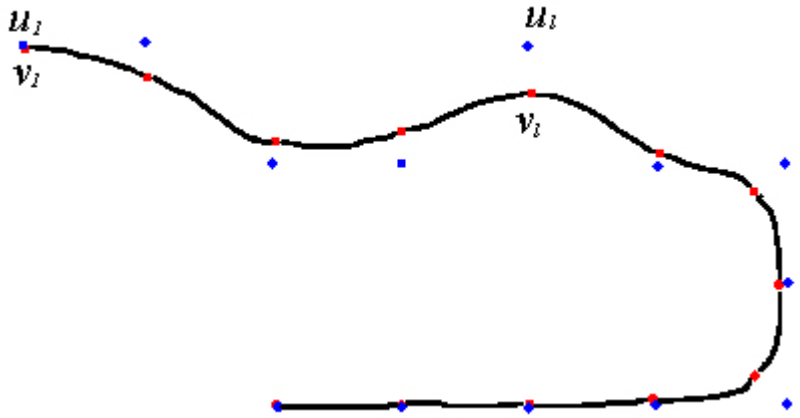


Figure 3.8 Illustration of the moving average filter

2. Cross section points computation

The cross section plane will be generated from the smoothed skeletal trajectory. Then four contour points will be computed in the cross section plane. These four contour points are distributed as a square in the cross section plane. Figure 3.8 illustrates the relationship between the centerline, cross section plane and contour points. The following describes the procedure to generate the cross section and four initial contour points:

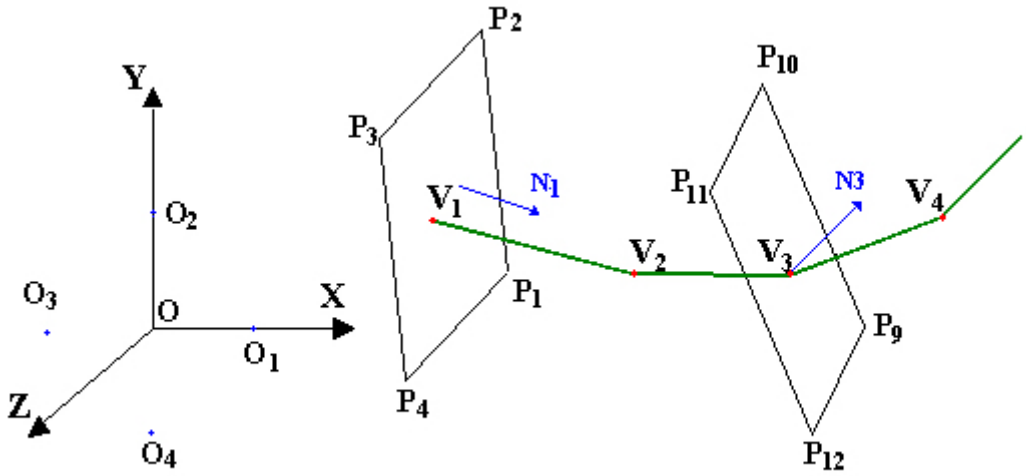


Figure 3.9 Generate the contour points in the cross section plane

In Figure 3.9, O is the world origin with $\vec{X}$, $\vec{Y}$ and $\vec{Z}$ coordinates, $\vec{V}_1$, $\vec{V}_2$, $\vec{V}_3$, and $\vec{V}_4$ are the centerline points in one of the vascular segment.

Step 1: Set the current centerline point C to the start centerline point of the vascular segment. Get the contour radius r at centerline point $\vec{V}_1$, set the four initial contour points $\vec{O}_1 = \{r, 0, 0\}$, $\vec{O}_2 = \{0, r, 0\}$, $\vec{O}_3 = \{-r, 0, 0\}$, $\vec{O}_4 = \{0, -r, 0\}$;

Step 2: Get the normal vector $\vec{N}$ of the cross section at current centerline point C. There are two cases to get the normal vector for different centerline point C:

1) If C is the start point or the end point (for example, V_1 in the figure), then the cross section is perpendicular to the centerline segment $\vec{V}_{21}$, so $\vec{N}_1 = \frac{\vec{V}_2 - \vec{V}_1}{\left\|\vec{V}_2 - \vec{V}_1\right\|}$;

2) If C is not at the start or end point (for example, $\vec{V}_3$ in the figure), then the cross section bisect the two successive centerline segments $\vec{V}_{32}$ and $\vec{V}_{43}$, thus $\vec{N}_3 = normalize(\frac{\vec{V}_3 - \vec{V}_2}{\left\|\vec{V}_3 - \vec{V}_2\right\|} + \frac{\vec{V}_4 - \vec{V}_3}{\left\|\vec{V}_4 - \vec{V}_3\right\|})$.

Step 3: Compute the four contour points at the cross section plane. We already have four initial contour points, to get the four contour points at the current cross section, we need to transform the initial four contour points. The transform matrix T contains a rotation angle θ and a translation vector $\vec{V}$:

For cross section at $\vec{V}_1$, $\theta = \angle(\vec{Z}, \vec{N}_1)$, $\vec{V} = \vec{V}_1$, so the four contour points at $\vec{V}_1$:

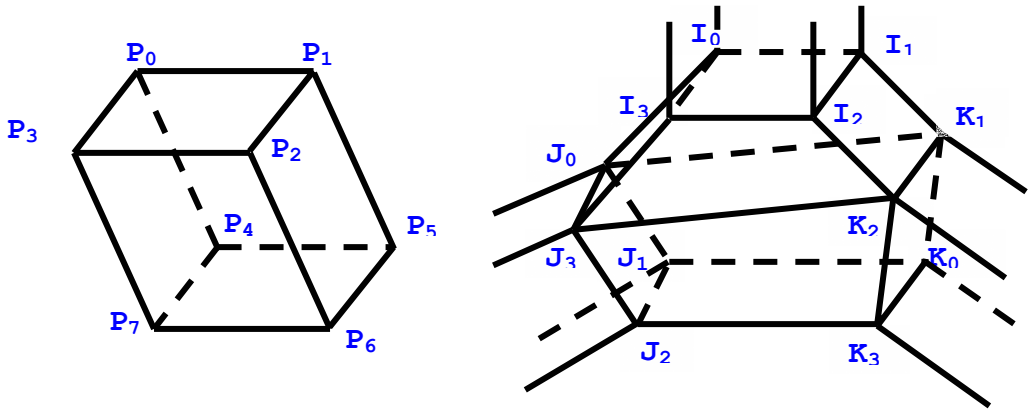
$\left\langle \vec{P}_1, \vec{P}_2, \vec{P}_3, \vec{P}_4 \right\rangle = T \times \left\langle \vec{O}_1, \vec{O}_2, \vec{O}_3, \vec{O}_4 \right\rangle$. With the same method, we can get the $\vec{P}_9, \vec{P}_{10}, \vec{P}_{11}, \vec{P}_{12}$

Step 4: Move the current centerline point to next node and continue with step 1 – 3, then get all the contour points of the vascular segment.

3. Mesh generation

After getting all the contour points of the vascular network, we connect them with a certain rule and get the control mesh. There are two rules for connecting contour points at adjacent cross sections.

Connection rule 1: This rule is adaptable to those cross sections that are located in the same vascular segment. The connection method is showed in Figure 3.10 (a).



(a) Connection rule 1 (b) Connection rule 2

Figure 3.10 Connection rules for generate control mesh

Following is the detailed description: {P_0, P_1, P_2, P_3} and {P_4, P_5, P_6, P_7} are two cross section adjacent. Firstly we find the matching points at adjacent cross sections (P_0

corresponds with P_4, P_1 corresponds with P_5, P_2 corresponds with P_6, P_3 corresponds with P_7), and connect them 2 by 2 (two points from the upper cross section, two corresponding points from lower cross section, for example, P_0 and P_1 correspond with P_4 and P_5), and then will get 4 surfaces:

- Surface1 $\{P_0, P_1, P_5, P_4\}$
- Surface2 $\{P_1, P_2, P_6, P_5\}$
- Surface3 $\{P_2, P_3, P_7, P_6\}$
- Surface4 $\{P_3, P_0, P_4, P_7\}$

Connection rule 2: This rule is adaptable to those cross sections that located in different vascular segments, i.e. the cross sections near a branching. Figure 3.10 (b) illustrates the connection method.

Compared with connection rule 1, this rule is more complex: Firstly, connect the four points of the parent cross section $\{I_0, I_1, I_2, I_3\}$ with the two points from the individual child cross section $\{J_0, J_1, J_2, J_3\}$ and $\{K_0, K_1, K_2, K_3\}$ to form 4 upper surfaces:

- Surface1: $\{I_0, I_1, K_1, J_0\}$
- Surface2: $\{I_1, I_2, K_2, K_1\}$
- Surface3: $\{I_2, I_3, J_3, K_2\}$
- Surface4: $\{I_3, I_0, J_0, J_3\}$

Then connect the contour points of two children cross section as if they are adjacent cross section described in connection rule 1, but exclude the surface formed by the four points used in previous step, i.e. the surface $\{J_0, J_3, K_2, K_1\}$. Then we get other 3 lower surfaces:

- Surface5: $\{J_0, K_1, K_0, J_1\}$

- Surface6: $\{J_1, K_0, K_3, J_2\}$

- Surface7: $\{J_2, K_3, K_2, J_3\}$

Using these two connecting methods recursively on the hierarchy vasculature structure, we can finally get the initial control mesh (the algorithm for generating the control mesh is showed below). As we can deduce from Figure 3.10, the generated initial control mesh will be very coarse: the branching parts of the control mesh will consist of 7 patches, while other parts will consist of 4 patches. If rendering the initial control mesh, we will get the rough contour surface of the blood vessel, especially at the parts of branching. For a vascular network that includes m line segments and n branchings, where each line segment contains $k_i, (i = 1,...,m)$ nodes, then the result control mesh will contain $4\sum_{1}^{m} k_i$ points and $(4\sum_{1}^{m}(k_i - 1) + 7n)$ faces.

Algorithm. Generating initial control mesh from vasculature hierarchy structure

Step 1. LineSegID = root; NodeID = 1;

Step 2. If NodeID has children, go to step 4;

Step 3. Connect the points in current cross section with the points in previous cross section according to the connecting method showed in Figure 3.10(a); then go back to step 2;

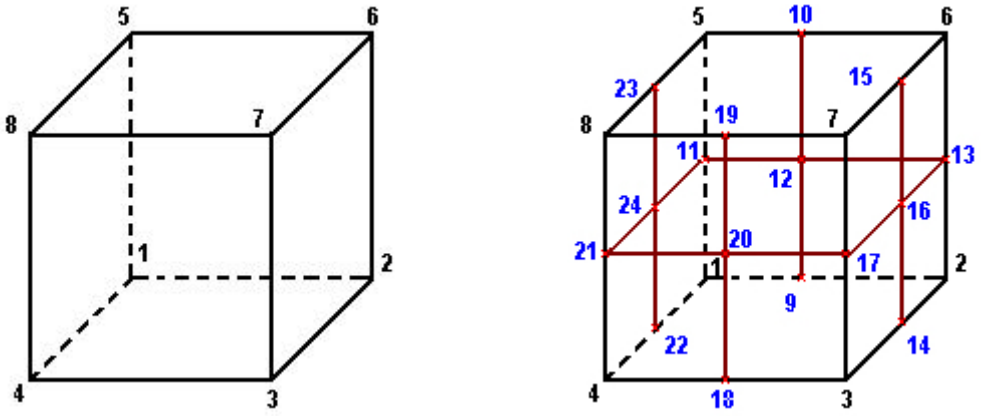
Step 4. Find the left child's LineSegID and right LineSegID; find the first cross section corresponding to the left child line segment and the right child line segment;

Step 5. Connect the points in current cross section with the points in left child cross section and right child cross section according to the connecting method showed in Figure 3.10(b);

Step 6. LineSegID = left child LineSegID; processing left child line segment as instruction above; LineSegID = right child LineSegID; processing right child line segment as instruction above.

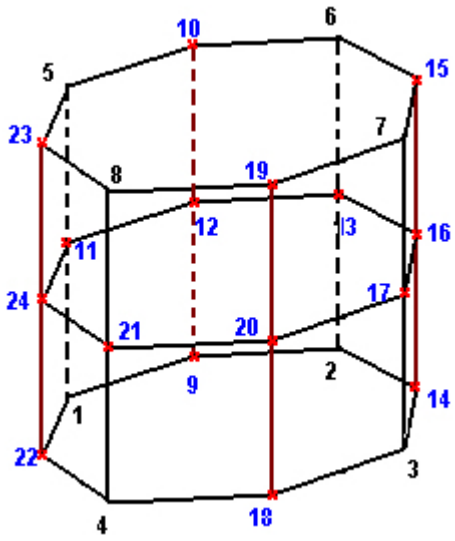
3.2.3.3 Surface subdivision

Surface subdivision can refine the coarse control mesh while keeping the contour unchanged. Using surface subdivision recursively, we can get the smooth mesh. There are different surface subdivision schemes, such as Loop [60], Catmull-Clark [61]. Both these two schemes can generate C^2-continuous surfaces (C^1-continuous surfaces at extraordinary vertices, for example, J_0 in Figure 3.10 (b)). Loop scheme applies to a triangular mesh while Catmull-Clark scheme applies to a quadrangle mesh. For the generated control mesh is quadrangular, we adopt a modified Catmull-Clark scheme. Figure 3.11 shows the process of the subdivision scheme.



(a) Initial mesh (8 vertices and 4 surfaces)

(b) Add new vertex and connect to form new surfaces



(c) Adjust the coordinates of the vertices

Figure 3.11 Illustration of the subdivision scheme

Step 1: First we have the initial mesh with 8 vertices {1,2,3,4,5,6,7,8} and 4 surfaces {{1,2,6,5}, {2,3,7,6}, {3,4,8,7}, {4,1,5,8}}, see Figure 3.11 (a);

Step 2: Get the center point of each edge and centroid point of each surface, this will add extra 16 vertices. Subdivide each surface into 4 subfaces and then get new 16 surfaces. Then calculate each vertex's valence (the number of surfaces that contains current vertex, for example, the valence of vertex 7 is 2 and vertex 20 is 4), see Figure 3.11 (b);

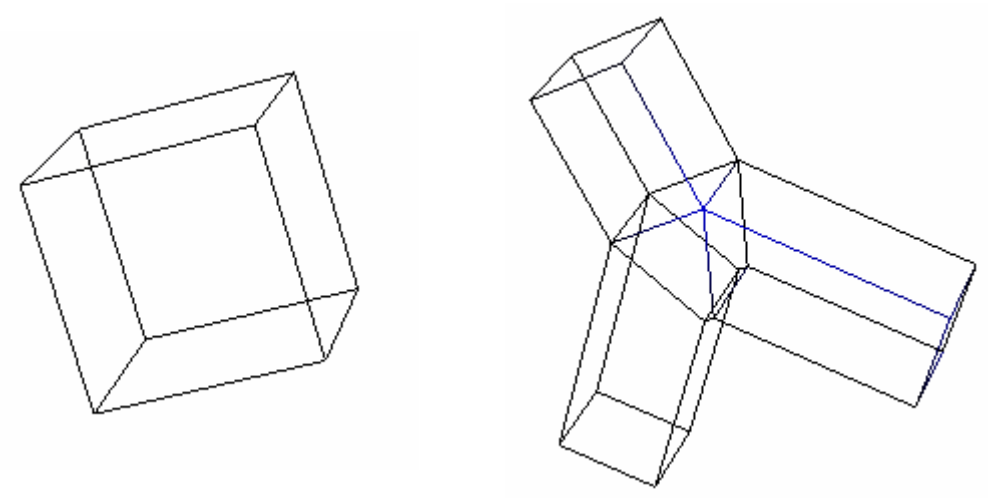
Step 3: Adjust the coordinates of each vertex. Assume the valence of vertex V is n, the surfaces that contain V are $S_i\ (i = 1,...,n)$, and then compute new coordinates of V as following:

$$V = \frac{1}{n}\sum_{1}^{n} centroid(S_i)$$

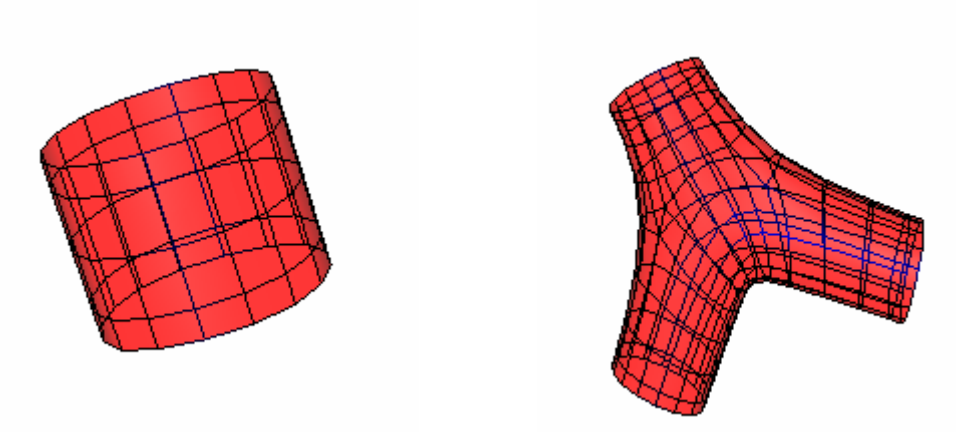
The result mesh is shown in Figure 3.11 (c), it is the first round subdivision.

Step 4: Apply step 1 – 3 to the first round subdivided mesh to get the second round subdivision mesh, and so on. From the above process, we can get that after subdivide one more times, the vertices are 3 times of previous vertices and the surfaces are 4 times of previous ones.

Figure 3.12 shows the resulting meshes after applying the subdivision scheme twice on the initial control meshes of type 1 and 2 individually. These two types' control mesh



(a) Initial control mesh, type 1 (b) Initial control mesh, type 2



(c) After subdivision, type 1 (d) After subdivision, type 2

Figure 3.12 Result of applying surface subdivision twice

are corresponding to the two connection rules in Figure 3.10. So after surface subdivision, the previous coarse vascular mesh will become smooth and the final vascular network will consist of meshes in Figure 3.12 (c) and (d).

3.3 Tools modeling

3.3.1 Representation of the tools

In the interventional neuroradiology procedures, tools include catheter and guidewire: A catheter is a thin and flexible hollow tube that is inserted into a bodily passage or cavity in order to allow fluids to pass into or out of it, to distend (expand) it, or to convey diagnostic or other instruments through it. It can be plastic or metal. A guidewire is a long and flexible fine spring used to introduce and position the catheter in a procedure. The guidewire can be made of steel or Titanium and in some cases they are coated with Teflon or a hydrophilic polymermaterial.

Compared with the human vascular networks, the tools are artificial and have simpler structure. We can represent them with their geometrical properties: the length and diameter of the tools. Although these tools have physical properties such as stiffness, torsion and friction, we assume them as rigid objects and only consider their geometrical model.

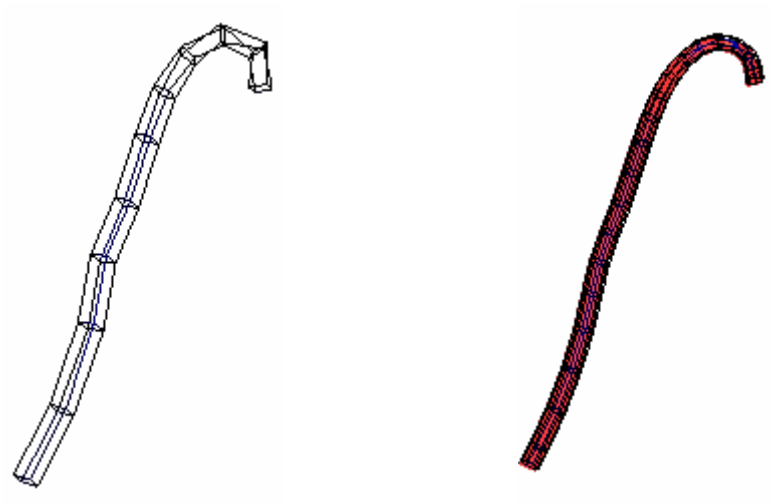
Following the same idea of vascular network representation, we get the central axis model of both the catheter and guidewire: their centerline points and the radius at each centerline point (the diameter is equal at every centerline point). Since the ratio of the diameter of the guidewire to the length of the tool is very small, it is possible to

approximate the guidewire as a string rather than a rod. This will simplify the model and expedite the computation of both collision detection and collision response.

3.3.2 Model reconstruction

Since the guidewire is simplified as a series of the line segment, we only discuss the model reconstruction of catheter.

Following the same idea of vasculature reconstruction, we get the centerline trajectory and radius at each centerline point. Then we generate the control mesh (Figure 3.13 (a)). Figure 3.13 (b) is the result of applying surface subdivision once to the initial mesh.



(a) Control mesh of catheter (b) Rendering after subdivision

Figure 3.13 Tools model

3.4 Discussion

In this section, two mainly problem encountered in the process of vasculature model are discussed: twisted cross section and vasculature diameter change.

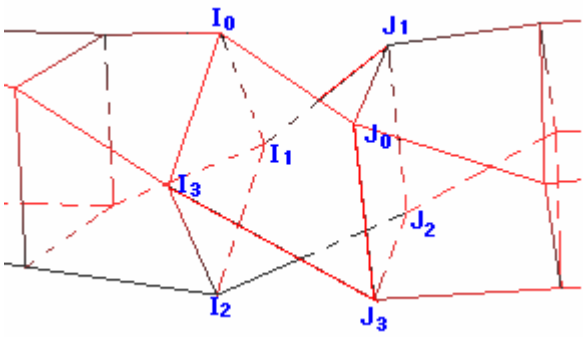
3.4.1 Twisted cross section

Twisted cross section occurs in the process of generating vascular control mesh. When we get the four contour points from the cross sections and connected them together according to the two connection rules described in section 3.2.3.2, the generated mesh seems to be twisted at some cross sections. Figure 3.14(a) shows this case: the control mesh begins twisted from the cross section I and reaches the most between cross section I and cross section J.

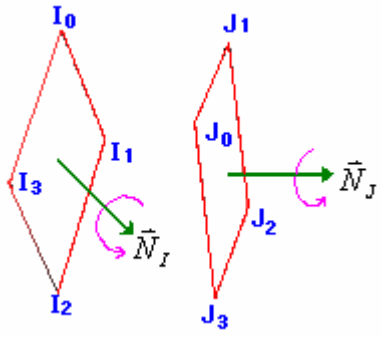
This problem arises for the direction of the vascular segment is changing always. Let's review the process of generating the four contour points (here take cross section I as example): First we calculate the coordinates of the initial four contour points in the world coordinates in XY plane; then we rotate the four points to the plane normal to vector $\vec{N}_I$ and translate to the centerline point; after that we get the final four contour points at the cross section I. As we can see in Figure 3.14(b), the final four contour points can rotate around the normal vector $\vec{N}_I$ and get various four contour points. If the adjacent four contour points are not mapped well, then the twist occurs.

Although the twist looks very sharp in the control mesh, the surface subdivision interpolate new vertices and surfaces and make the twisted part look smooth. From

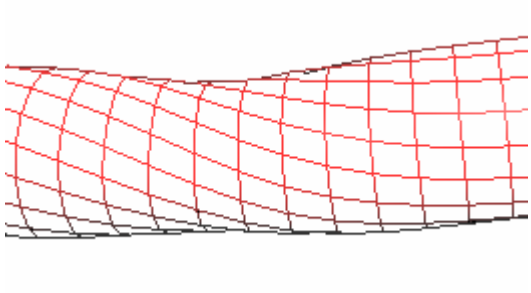
Figure 3.14 (c) and (d), we can see that the subdivided mesh and rendering result of the twisted part become smooth. It is satisfactory for the web-based simulation system.



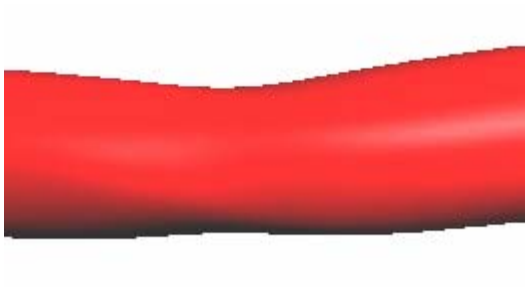
(a) Twist occurs at cross section I and J



(b) Contour points rotation



(c) Subdivision at twisted part



(c) Rendering result at twisted part

Figure 3.14 Twisted cross section problem

3.4.2 Vascular model diameter change

In the process of surface subdivision, the coordinates of the vertices will be adjusted in each subdivision round. This adjustment will make the diameter of the vascular model a little changed. As we can see from Figure 3.14 (a) and (c), the diameter of the control mesh is a little bigger than the subdivided mesh. The ratio of the final diameter to the initial diameter changes with the shape of the vasculature and the number of the surfaces in the subdividing mesh. It is hard to give a formula to describe the change ratio in mathematics.

Although the diameter changes a little from the control mesh to the subdivided mesh, the shape and the direction of the vascular model is preserved. With regard to the central axis model is a simplified one of the real vascular network and the ratio of the diameter variation to the length of the vascular is very small, it is acceptable to allow the diameter change in the model.

Chapter 4

Haptic modeling

The goal of haptic rendering is to enable the user to touch, feel and manipulate virtual objects through a haptic interface. In our web-based medical simulation system, the force feedback is widely assumed to enhance the performance of the trainee. Without the force feedback, the trainee will not be aware that the tools collide with the blood vessel wall and he/she should stop pushing the tools forward.

In this chapter, modeling the haptic feedback is discussed in section 4.1. In section 4.2, how to render the haptic force feedback and the behaviour of the tools are described. Collision detection, collision response and tools' behaviour rendering are also discussed in this section.

4.1 Haptic modeling in the simulation procedure

4.1.1 Haptic feedback in simulation

The interventional neuroradiology procedure often involves inserting the tools into the blood vessels and unblocking the artery to restore the blood flow. In the catheterization procedure, there exist mainly two types of force feedback that can be passed to the trainee:

1. The resistance from the blood flows. For the blood is a sticky fluid and flows at a certain speed, when the trainee moves the tools forward or rotates the tools, the blood can restrain the motion of the tools and give the trainee a feel of resistance.

2. The resistance from the contact between the blood vessel and the tools. When the tools collide the blood vessel wall, it will be hindered not only by the blood flows but also the friction and bounce force from the blood vessel wall. If the tools are approximately perpendicular to the blood vessel wall or blocked by the complex vessel, then this resistance can stop the motion of the tools. Under such cases, the trainee can only pull the tools back or rotate the tip of the tools and change the moving direction.

4.1.2 Model the force feedback

Compared with the 2nd type resistance, the first type resistance is rather small and can be ignored during the haptic modeling. For the haptic rendering is done in the web server side, it is reasonable to simplify the force feedback computation model and only focus on the 2nd type resistance.

To simplify the haptic model of the 2nd type resistance, we assume the tools and the blood vessel as rigid objects, i.e. they will respond to the force feedback but will not deform. In our haptic model, the blood vessels are modeled as surface without thickness and the tools as a series of line segments. The tools are not permitted to penetrate through the blood vessel wall.

Some physical properties are also introduced into the haptic model to compute the force feedback, such as friction coefficient, tools moving speed.

4.2 Haptic rendering

The haptic rendering includes two parts: one is the force feedback from the collision response, the other is the behaviour rendering of the tools, i.e., how the catheter and guidewire respond when a collision occurs.

4.2.1 Collision detection

Collision detection plays an important role in the haptic rendering. In our web-based medical simulation system, the objective of the collision can be summarized as: finding the occurrence and location of the collision points between the blood vessel and catheter/guidewire. Only when the collision is detected can the collision response and behaviour of the catheter/guidewire be calculated.

For the final vascular mesh model contains thousands of quadrangles and the haptic rendering part is done at the server side, the computation of collision detection between the vascular model and the catheter/guidewire model should be completed within millisecond over the Internet. So fast collision detection algorithm is necessary for real time web-based haptic rendering.

We adopted the fast and accurate collision detection algorithm described in [62]. It pre-computes a hybrid hierarchical representation, which consists of uniform grids and

trees of tight-fitting oriented bounding box trees (OBB tree) [63]. The hierarchical representation supports fast proximity query and is applicable for collision detection between catheter/guidewire and the blood vessel. Figure 4.1 illustrates the building procedure of the hierarchical OBB tree and how the OBB tree is used to fast detect the collision between the vascular model and catheter/guidewire.

Suppose we have a vascular model shown in the first layer *I* (Figure 4.1), the arrow in the first layer indicates where the catheter is currently located. The collision detection procedure can be divided into two steps:

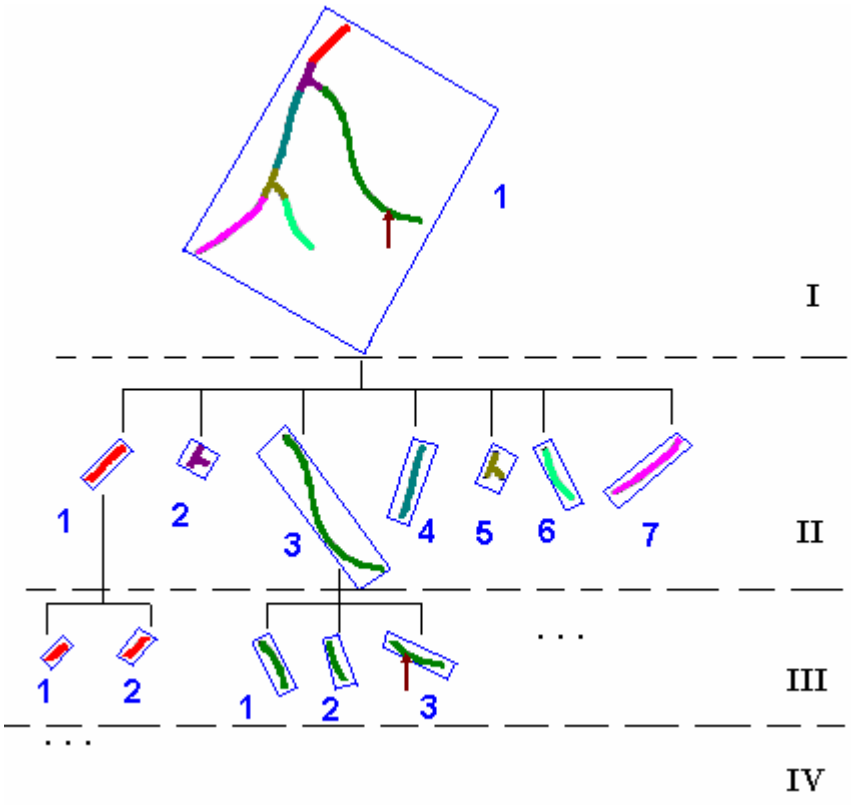


Figure 4.1 OBB tree for collision detection (in 2D view)

1. OBB tree building

In this step, we build a hierarchical OBB tree from top to bottom. At the first layer, we calculate the oriented bounding box of the whole vascular model. Then we divide the whole vascular model into 7 parts according to the central axis model, where 2 branches (2 and 5) and 5 vascular segments (1,3,4,6,7), calculate the 7 vascular parts' oriented bounding boxes individually and then divide the 7 parts again to generate next layer, and so on. Finally a hierarchical structure that contains OBBs is build, the root node of the OBB tree is the OBB of the whole vascular network and the leaf node is every surface that consists of the vascular mesh.

2. Collision query

The collision query is to search the whole OBB tree and detect in which bounding box the catheter/guidewire is located. If the catheter/guidewire is found in one OBB, then check its sub-OBB and see in which sub-OBB the tools are located. The query will end when it reaches the leaf node, i.e., the surface that consists of the vascular mesh. Let's give an example to describe the detail: in Figure 4.1, the catheter is located at the position where the arrow indicates, and then we will detect whether the tools collide with the vascular model and where the collision takes place.

The collision query begins at the root node of the OBB tree, i.e., the OBB 1 at layer *I*. Apparently the catheter is located in the OBB 1 at layer *I*, then the query will be stepped to the 7 children nodes of the root node at layer *II*. The catheter will be found in OBB 3 at layer *II*, querying the tree recursively and then will be found in OBB 3 at layer *III*, and so on.

As we can get from the searching procedure, the computation complexity of the query time is $O(\log N + m)$, where N is the vascular segments number and m is the average surface number of each subdivided vascular node.

4.2.2 Collision response

When a collision occurs, we calculate the force feedback according to the scene (the moving velocity of the catheter/guidewire and the angle between the catheter/guidewire and the blood vessel wall, etc.), and the force will be passed to the trainee via a force feedback device [64, 65].

Figure 4.2 shows the force feedback computation when catheter/guidewire collides with the blood vessel wall: Suppose the guidewire collides the vessel wall at some point, the angle between the guidwire and the normal vector $\vec{N}$ of the surface is θ. Two types of forces are generated at the contact point: the friction (F_1) and bounce force (F_2), the force passed to the guidewire, F, is the sum of the two forces. F1, F2 and F are coplanar.

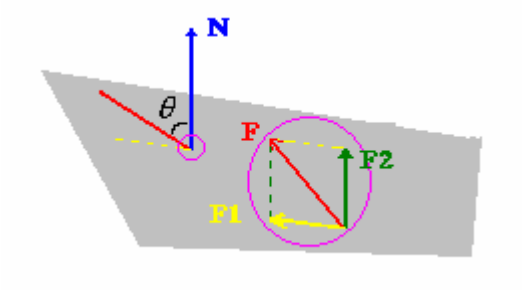


Figure 4.2 Force feedback computations at contact point

4.2.3 Tools' behaviour

Under normal cases, after the trainee push the tools forward, they will follow the direction of the tip and step ahead. When a collision occurs, although the force feedback will give the trainee a resistance, the catheter/guidewire will still move forward unless they are blocked by the vasculature. Under such case, the direction of the catheter/guidewire tip will change and the catheter/guidewire will move forward according to the new tip direction. Figure 4.3 illustrates the direction change of the tip when a collision occurs.

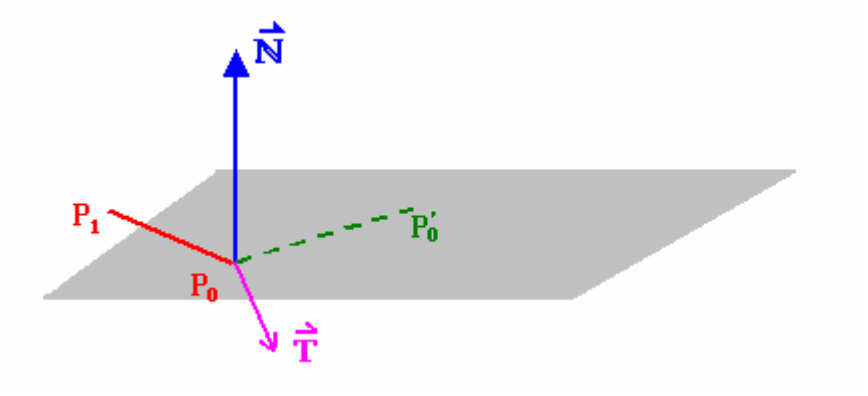


Figure 4.3 Forward direction of the tool

Suppose $\vec{N}$ is the normal vector of the surface, P_0 is the contact point and $\overrightarrow{P_0P_1}$ is the tip direction of the tool. We will calculate the direction of the new tip direction $\overrightarrow{P_0P'_0}$. As we know, after the collision, the tip direction will change and follow the surface of the blood vessel wall, which means that $\overrightarrow{P_0P_1}$, $\vec{N}$ and $\overrightarrow{P_0P'_0}$ are coplanar, so we calculate the vector $\vec{T}$ that normal to the plane $P_1P_0P'_0$:

$$\vec{T} = \overrightarrow{P_1P_0} \times \vec{N}$$

For $\vec{T}$ is the normal vector of plane $P_1P_0P'_0$, so:

$$\overrightarrow{P_0P'_0} = \vec{N} \times \vec{T}$$

4.3 Discussion

In the modeling of haptic feedback in the simulation procedure, we used a simplified haptic model and treat the vascular model and tools model as rigid objects. Although these simplifications will lower the accuracy of the haptic feedback in the simulation system that provides to the trainee, it's acceptable for a web-based application system. After all, realism is more important than accuracy in such a virtual environment. In the real procedure, the magnitude of the force feedback is very small and the trainee often uses the force feedback as a guide to navigate the tools in the vasculature. In our web-based simulation system, the visual and haptic feeling together provide the navigation functionality to the trainee. The visualized vascular network and virtual tools can let the trainee know where the tools are located; the haptic feedback let the trainee know when to stop moving the tools ahead, rotate the tools and change the moving direction or pull the tools back.

Chapter 5

System Implementation

In this chapter, the details of system implementation are discussed. Section 5.1 introduces all the modules in our system. The environment for web server side and client side are briefly discussed in section 5.2. System interface of the web-based system are showed in section 5.4. The last section is about some technical evaluations of the whole system.

5.1 System architecture

A typical web-based system is divided into three tiers: (1) presentation layer, (2) business logic layer, and (3) database layer [8]. The proposed system has a 3-tier architecture as shown in Figure 5.1.

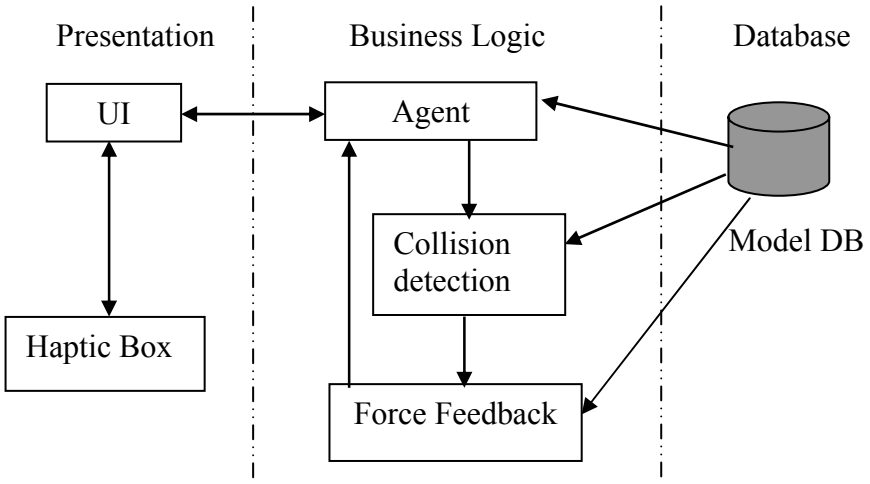


Figure 5.1 Architecture of the system

Within this architecture, each part has different functionality:

1) UI – it is a Java3d-enabled browser, whose main function is to visualize the vascular and tool model. It is the interface that the trainee interacts with the virtual environments. It also transmits the tracking signal of the simulator to the agent and sends the control signal from the agent to the simulator.

2) Haptic Box – it is both a simulator with catheter/guidewire and force feedback hardware. As a simulator, it traces the trainee's operations (motion and rotation) on catheter/guidewire; the UI collects these operation signals and update the virtual catheter/guidewire in the virtual environments. As a force feedback device, it can receive the control signal from UI and give the trainee a force feedback. Figure 5.2 shows a prototype of this haptic box.

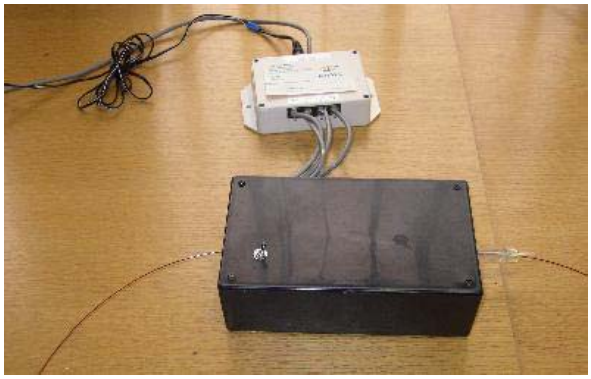


Figure 5.2 Simulator prototype device with force feedback

3) Agent – it acts as an intermediate between client and server. Its main functions include transferring vasculature and catheter/guidewrie model data to the UI and communicating between UI and collision detection module and force feedback module.

4) Collision Detection – it is a software module which resides at the server side and is used to check whether the catheter/guidewire contact with the vascular model. If a

collision is detected, it sends the information about the collision (the angle, the friction coefficient at the contact point, etc.) to the force feedback module introduced below.

5) Force Feedback – it is also a server side software module. If a collision occurs, then it will calculate the magnitude of resistance force feedback according to the information of the collision transmitted by the collision detection module.

6) Model DB – it is a database that stores the models for training.

Although the architecture gives us a clear profile about the system, there are still several implementation details which need to be clarified.

5.1.1 Model

The central axis model used in the system is a compact representation of the vascular network and is small in data size. To speed up the transmission from the server to the browser, data compression is also used in our system. Compared with the transmission time over Internet, the time for uncompressing the model data is very small; this method is useful when the network is not stable.

5.1.2 Visualization

With regard to the different platforms used at the client side, our system uses Java and Java3D to implement the whole system. In addition, Java3D delivers good 3D graphics rendering performance over different platform. The small data size of the central axis model guarantees no-delay changes over the Internet, and Java and Java3D guarantee a

universal UI over the Internet. The system can be easily migrated from one platform to another.

5.1.3 Collision Detection and Force Feedback

As the kernel parts at the server side, collision detection module and force feedback module play the most important role in the system. To provide the trainee a realistic training environment, the system needs to give the trainee real-time collision detection and force feedback. However, in some other simulation systems, a sound is played when a collision occurs, this method cannot give the trainee experience on how to navigate the catheter/guidewire in human vasculature during the neuroradiology procedure.

Collision detection and force feedback can be composed of several CORBA servers. They may reside at one machine or are distributed over several machines. This distribution guarantees real-time response when the trainees do continuous actions. The computation workload is distributed over different CORBA servers even when several trainees exercise simulation simultaneously in the system.

5.2 System environment

As a web-based application, there are different environment needed at the server side and client side.

The server side needs a web server that can be accessed by the users over the Internet; all the software modules are also running at the server side. The collision detection module and force feedback module can run at several machines simultaneously.

Compared with environment at the server side, the client side is much simpler. It only needs a Java3D-enabled web browser, a haptic box connected to the machine with the software driver correctly installed. When the client browser begins to access the web-based simulation system, the UI applet will be downloaded from the server to the client side and launched to communicate with the agent module at server side and the local haptic box.

5.3 System interface

Figure 5.3 shows the UI which is seen in the client browser, the interface divides the whole client browser window into 3 areas: the left is the global view of the whole simulation system, both the vascular model and catheter/guidewire can been seen in global view. The vascular model can be switched between wire-frame mode and surface rendering mode (in Figure 5.3, the vascular model is in wire-frame mode). The

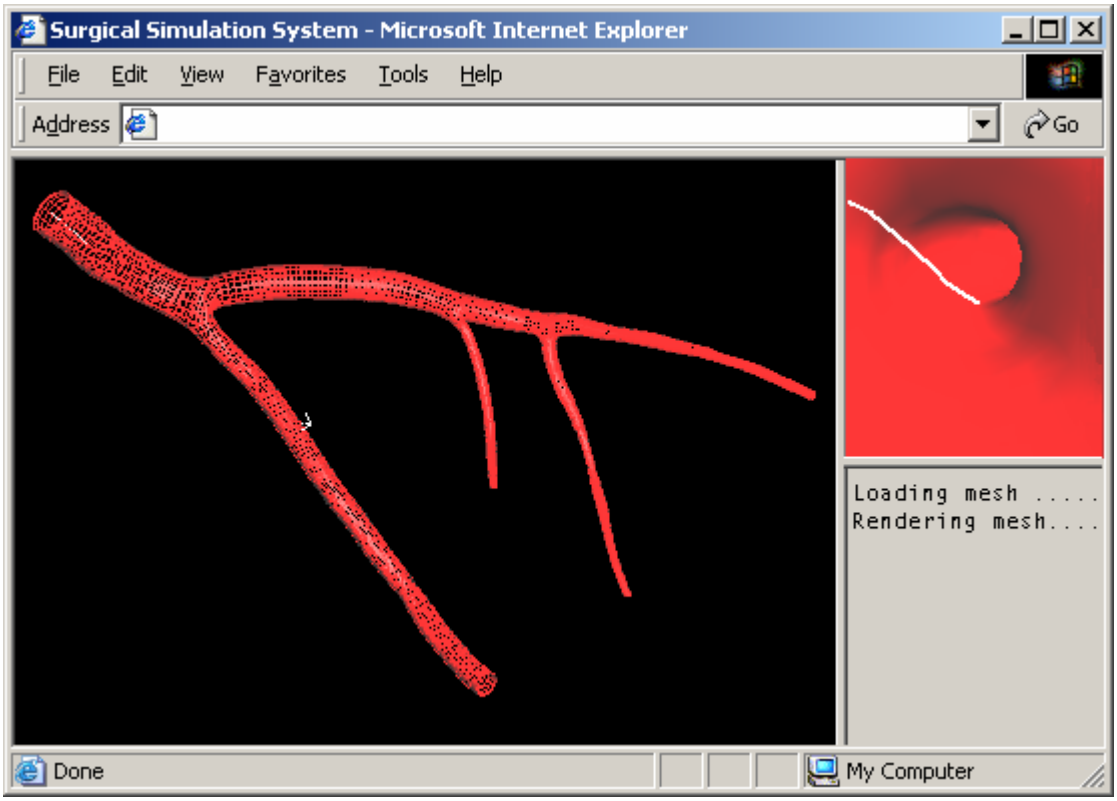


Figure 5.3 The UI seen from the client side (in browser)

user can use the mouse to zoom in/out in the global view and translate its position. The upper right area is the local view that is seen inside the vasculature; this view will change with the motion of the catheter/guidewire. The bottom right area is the output window, system messages can be shown in the output textbox. Collision information will also be given in this window when no haptic box is connected to the system.

Figure 5.4 and Figure 5.5 are the enlarged local views occurred at the non-bifurcation

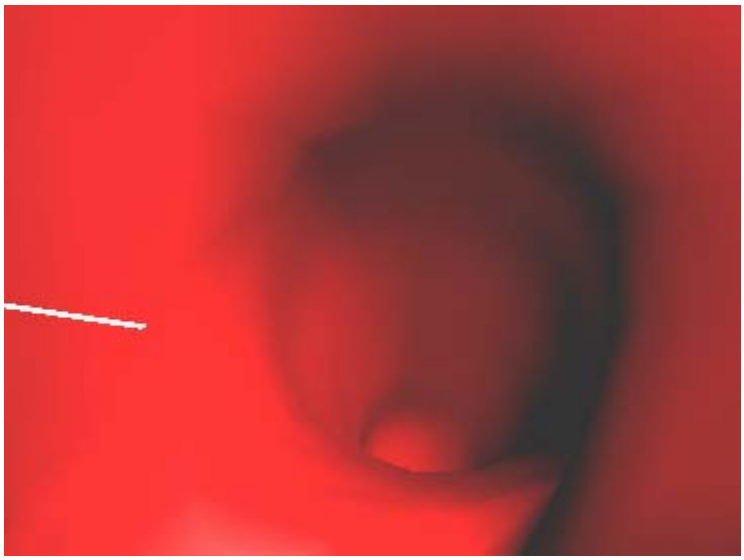


Figure 5.4 Local view of the non-bifurcation part

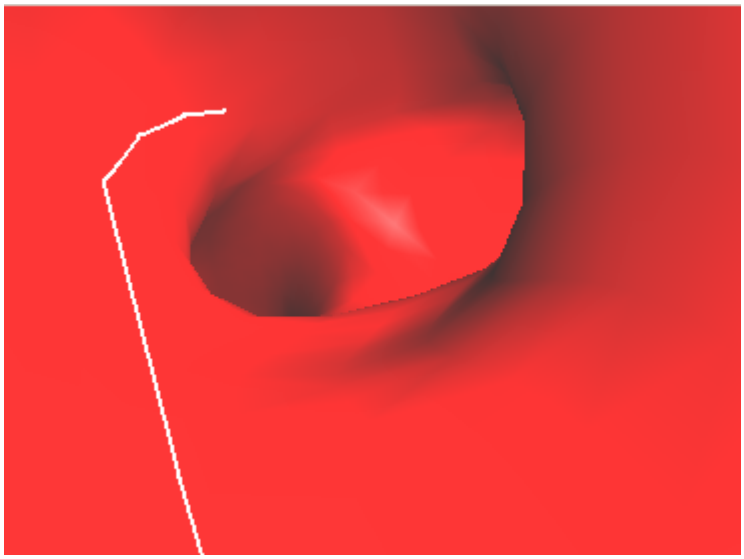


Figure 5.5 Local view of the bifurcation part

part and bifurcation part. As we can see from the two figures, the surface inside the blood vessel is also smooth both at the normal part and the branching part.

5.4 Evaluation

In this section, some technical evaluations about the whole system are given. The testing environment is as follows:

CPU: Pentium III 600 MHz
Memory: 512M RAM
Video: 32M RAM
OS: WIN2000 professional

As a real-time medical simulation system, the graphical frame in our system is about 24.9 frames/seconds, and the haptic box's rate is 1K Hz. For a vascular mesh that contains about 2,600 quadrangles, the collision query time is less than 0.05s. All these arguments can meet the requirement of a real-time system.

Table 1 compares the model data size and transmission time before and after data compression. The time needed after compression is less than one minute, which can meet the acceptance of the user [66].

Table 1. File size and time consumption before/after compression

Type	File size - before	File size – after	Time - before	Time - after
Aneurysm	362KB	50KB	30s	5s
Abdomen	257KB	41KB	25s	4s
Heart	30KB	4KB	3s	1s
Phantom	151KB	20KB	14s	2s
Human body	2MB	135KB	190s	14s

Chapter 6

Conclusion and future work

6.1 Conclusion

The purpose of this dissertation is to develop a web-enabled medical simulation system for interventional neuroradiology procedures. This system can provide the trainee a high fidelity virtual environment in both visualization and haptic rendering. As an excellent media for information delivery, WWW offers accessibility and distributed computing which can be used to provide novel solutions for traditional applications. Combined with Java3D, the web-based medical simulation system is independent of platform and also has scalability.

The physical-based modeling of the vascular network which is proposed in this work is an important feature. In this model, the central axis model is used to represent the human vasculature and a simplified control mesh is reconstructed for 3D visualization. The succeeding surface subdivision makes the coarse control mesh smooth and the visualization after rendering is satisfactory for the simulation system. The data structure to store the axis model is very small, which shortens the transmission time from the web server side to the training client side. The hierarchical structure of the model is also convenient to build the Oriented Bounding Box (OBB) tree for fast collision detection.

Haptic rendering in the web-based simulation system is another important feature. Although the system assumes the vasculature model and tools' model as rigid objects,

it provides the trainee a relatively realistic force feedback environment over the Internet. The compact data structure and the fast collision detection algorithm make the haptic rendering a real-time one. The distributed haptic force feedback computation model in our system also let the client side focus on the quality and performance of 3D model visualization.

6.2 Future work

Our system provides a good start in modeling and haptic rendering of the medical procedure over the Internet. However, there is still some spaces left to be done in the future.

Deformable object modeling and FEM can be introduced to the system and make the simulation procedure more realistic. In our current implementation, the vascular model and tools are treated as rigid objects and the force feedback is a linear one. After the physical properties of the vasculature and tools are introduced to the simulation system, such as the elasticity property and stiffness, the deformation procedure of the blood vessel wall and the tools can be visualized during the training procedure. FEM can also calculate the non-linear force feedback occurred when the tools collide with the blood vessel wall. Both methods can enhance the reality of the system.

Pathological vasculature modeling is another field to enhance the functionality of the simulation system. Aneurysm and stenosis are the typical pathologies in human vasculature. Different patients have different pathological vasculature structures. Therefore it is a challenging task to model the pathologic part for different patients.

And what's more, the structure of the pathological parts is more complex than that of the normal parts. If the pathological vasculature model is applied in the system, the performance of the training procedure will be increased a lot.

The last space for enhancement is the evaluation and validation of the system. In our system, only limited validation and performance evaluation have been done, such as the graphics frame rate, haptic response rate and transmission time. As one of the three important criteria to evaluate a medical simulator, complete evaluation and validation from the trainee and medical staff, such as the correctness of model and the realism of force feedback are necessary for an application system.

Appendix A

HCLM File

A.1 HCLM file format

The segmented vascular data are stored in HCLM files. It is a text file that organizes vascular information according to the Central Axis Model. The following data shows the file data format.

```
s 4
g 99
l main
f 60.000000
p 0
c 1 100
n 21
v -6.673064 12.703399 -47.877568 4.990000
v -7.873580 11.309175 -44.813852 4.940000
v -9.074098 9.914953 -41.750136 4.890000
… …
```

In the HCLM file, each line of the data begins with a label as one characteristic of the vascular information. Following the label is the corresponding data information. In the above example, the first line “s 4” means that there are 4 segments in the vascular network. Next is the description for each segment. “g” labels the ID of the segment; “l” labels the name of the segment; “f” labels the blood flow in the vessel; “p” labels the parent information of the segment in the vascular network, subsequent data are the number of parent segment and IDs of its parent segment; “c” labels the child information of the segment in the vascular network, similar to “p”; “n” labels the number of vertexes in the segment; “v” labels the vertex information, subsequent data

are the coordinates of each vertex. All the segments are described in this format one bye one.

A.2 Example data

Following is a complete hclm file of the catheter.

```
s 1

g 0
l test
f 0.0
p 0
c 0
n 11
v 1.20 0.57 0.0 0.03
v 1.18 0.48 0.0 0.03
v 1.09 0.46 0.0 0.03
v 1.02 0.53 0.0 0.03
v 0.98 0.63 0.0 0.03
v 0.96 0.75 0.0 0.03
v 0.94 0.91 0.0 0.03
v 0.91 1.05 0.0 0.03
v 0.91 1.21 0.0 0.03
v 0.88 1.41 0.0 0.03
v 0.83 1.58 0.0 0.03
```

Author's Publication

1. Yiping Lu, Xin Ma, CheeKong Chui, KiaFock Loe, and Wieslaw L. Nowinski, *Web-based Simulation System for Interventional Neuroradiology Procedures*, Proceeding for International Congress on Biological and Medical Engineering, 2002
2. Xin Ma, Yiping Lu, KiaFock Loe, and Wieslaw L. Nowinski, *Haptic device for web-based training system for interventional radiology procedures,* Computer Assisted Radiology and Surgery 2003
3. Yiping Lu, Xin Ma, KiaFock Loe, CheeKong Chui, and Wieslaw L. Nowinski, *Haptic Vascular Modeling and Visualization in Web-enabled Interventional Neuroradiology Simulation System*. Accepted in Medical Imaging 2004.
4. Xin Ma, Yiping Lu, KiaFock Loe, and Wieslaw L. Nowinski, *Haptic Interface of Web-based Training system for Interventional Radiology Procedures,* Accepted in Medical Imaging 2004.

Reference

[1] N.H. Khalili, K. Brodlie and D. Kessel, *WebSTer: A Web-based Surgical Training System*, Medicine Meets Virtual Reality, pp. 69-75, IOS Press. 2000

[2] N. John, N. Phillips, R. Vawda, and J. Perrin, *A VRML Simulator for Ventricular Catheterization,* Proc of the Eurographics UK conference, April 1999

[3] G.A. Higgins et.al, *New Simulation Technologies for Surgical Training and Certification: Current Status and Future Projects,* Presence, 6(2): 160-172, 1997.

[4] W.L. Nowinski, A. Fang, B.T. Nguyen et al, *Multiple Brain Atlas Database and Atlas-based Neuroimaing System,* Computer Aided Surgery, Vol. 2, No. 1, pp. 42-66, 1997

[5] Y. Wang, C.K. Chui, H.L. Lim et al, *Real-time Interactive Simulator for Percutaneous Coronary Revascularization Procudures*. Computer Aided Surgery, Vol. 3, No 5, pp.211-227, 1998

[6] J. Anderson, R. Raghavan, Y. Wang, and C.K. Chui, *daVinci – A Vascular Catheterization Simulator, Journal of Vascular and Interventional Radiology*, Vol. 8, No. 1, pp. 261, 1997

[7] Y. Lu, X. Ma, C. K. Chui, K. F. Loe, and W. L. Nowinski, *Web-based Simulation System for Interventional Neuroradiology Procedures,* Proceeding for International Congress on Biological and Medical Engineering, Dec 2002

[8] S.L. Dawson SL, J.A. Kaufman, *The Imperative for Medical Simulation*, Proceedings of the IEEE Vol. 86 No. 3, pp. 479-483, 1998

[9] R.M. Satava, S.B. Jones, *Medical applications of virtual reality*, Handbook of Virtual Environments: Design, Implementation, and Applications. Mahwah, NJ: Lawrence Erlbaum Associates Inc, pp.368-391. 2002

[10] G. Riva, *Virtual reality for health care: the status of research*, Cyberpsychology & Behavior, 5 (3): 219-225, 2002

[11] G. Riva, *Applications of Virtual Reality in Medicine*, Methods of Information in Medicine, 5 (5), pp. 524-534, Oct 2003

[12] C. Chinnock, *Virtual reality in surgery and medicine,* Hosp Technol Ser 1994, Vol. 13 (18), pp1-48. 1994

[13] T.V. Jakob, *A state of the art review of methods and applications in medical Virtual Reality*. Virtual Reality Meets Medicine, 2001

[14] R.M. Satava, *Surgical education and medical simulation,* World Journal of Surgery, 25 (11): pp1484-1489. 2001

[15] R.M. Satava *Surgery 2001: A Technologic Framework for the Future,* Surgical Endoscopy 7: pp111-113. 1993

[16] R.M. Satava, *Virtual reality surgical simulator: The first steps,* Surgical Endoscopy 7 (3): pp203-5. 1993

[17] X.S. Chen, X. Ma, Y.Y. Cai, C.K. Chui, Y.P. Wang, W.L Nowinski, *Modeling and Visualization of Vasculature Interventional Neuroradiology Simulation*, the 7th international conference on computer aided design and graphics, 2001

[18] N.H. El-Khalili, *Surgical Training on the World Wide Web*, PhD thesis, University of Leeds, United Kingdom. 1999

[19] E. Hciler and H. Breltwieser, *Telepresence systems for application in minimally invasive surgery*, In Medicine Meets Virtual Reality II, pp 77-9, San Diego, January 1994

[20] U. Kuhnapfel, *Realtime graphics computer simulation for endoscopic surgery*, In Medicine Meets Virtual Reality II, pp 114-16, San Diego, January 1994

[21] D. Meglan, R. Raju, and et.al, *The Teleos virtual environment toolkit for simulation-based surgical education*, In S. Weghorst H. Sieburg and K. Morgan, editors, Health Care in the Information Age, pp346-351. IOS Press and Ohmsha, 1996

[22] J.R. Merril, G.L. Merril, and et.al, *Photorealistic interactive three-dimensional graphics in medical simulation*, In K. Morgan and R. Satava, editors, Interactive Technology and New Paradigm for Healthacre, pp244-52. IOS Press and Ohmsha, 1995

[23] C.K. Chui, Z. Li, J.H. Anderson, K. Murphy, A. Venbrux, X. Ma, Z. Wang, P. Gailloud, Y. Cai, Y.Wang and W.L. Nowinski, *Training and Planning of Interventional Neuroradiology Procedures - Initial Clinical Validation*, Vol. 85, pp. 96-102, Medicine Meets Virtual Reality, 2002

[24] J. Anderson J, C.K. Chui, Y.Y. Cai, Y.P. Wang, Z.R. Li, X. Ma, W.L. Nowinski, et al, *Virtual reality training in interventional radiology: the Johns Hopkins and Kent Ridge Digital Laboratory Experience*, Seminars in Interventional Radiology 19(2): 179-185, 2002

[25] M.R. Ali, Y. Mowery, B. Kaplan, E.J. DeMaria. *Training the novice in laparoscopy*, 8th World Congress of Endoscopic Surgery, 16(12): 1732-6, 2002

[26] M. Gor, R. McCloy, R. Stone, A. Smith, *Virtual reality laparoscopic simulator for assessment in gynaecology,* 110 (2): 181-187, Bjog 2003

[27] M. Alcañiz, C. Perpiña, R. Baños, J.A Lozano, M.D. Ross, T.A. Twombly, C. Bruyns, R. Cheng, S. Senger, *Telecommunications for health care over distance: the virtual collaborative clinic*, Studies in Health Technology and Informatics 2000, 70: 286-91, 2000

[28] W. Doyle, *Interactive image-directed epilepsy surgery: Rudimentary virtual reality in neurosurgery*. In K. Morgan and R. Satava, editors, Interactive Technology and New Paradigm for Healthacre, pp91-100. IOS Press and Ohmsha, 1995

[29] W. Bauer, H. Bullinger, and et.al. *Planning of orthopaedic surgery in virtual environments by the example of osteotomy operations,* In K. Morgan, R. Satava and H. Sieburg, editors, Interactive Technology and the New Paradigm for Healthcare, chapter 5, pp29-36. IOS Press and Ohmsha, 1995

[30] N.W. John, M. Riding, *Surgical Simulators on the World Wide Web – This Must be the Way Forward?* Proceedings of UKVRSIG 99, University of Salford, September 1999

[31] N.I. Phillips, N.W. John, *Web-based Surgical Simulation for Ventricular Catheterisation*, Neurosurgery Vol. 46 No. 4, pp933-937, April 2000

[32] N.W. John, N.I. Phillips, *Surgical Simulators Using the WWW*, Medicine Meets Virtual Reality 2000, Newport Beach, California, January 2000.

[33] Advanced Surgical Planning - Interactive Research Environment. http://wwwmed.stanford.edu/aspire/

[34] N. Brooke, Steele, Joy P. Ku, and et.al. *Internet Based User Interface For Computer Aided Surgical Planning,* ASME Summer Bioengineering Conference, BED, Vol 42, pages 21-22, 1999

[35] A. Dodd, M. Riding, N.W. John, *Building Realistic Virtual Environments using Java and VRML,* In 3rd Irish Workshop on Computer Graphics, March 2002

[36] P.P. Anna, *Cerebral Blood Vessels Modelling*, Polytechnical University of Catalunya, LSI-98-21-R, Spain 1998

[37] P. Suetens, et al, *Image Segmentation: Methods and Application in Diagnostic Radiology and Nuclear Medicine*, European Journal of Radiology, vol.17, pp.14-21,1993.

[38] L.P Clarke, et al, *Review of MRI segmentation: methods and applications, Magnetic Resonance Imaging*, vol. 13,pp. 343-368, 1995

[39] H.C. Cline, W.E. Lorensen, R. Kikinis and F. Jolesz, *Three-dimensional segmentation of MR images of the head using probability and connectivity,* Journal of Computer Assisted Tomography, 14(76):1037-1045, 1990

[40] H.H. Ehricke, K. Donner, W. Koller, and W. Straber, *Visualization of vasculature from volume data,* Computer and Graphics, 18(3):395-406, 1994

[41] R. Verbeeck, D. Vandermeulen, J. Michiels, P. Suetens, G. Marcha, J. Gybels, and B. Nuttin, *Computer assited stereotactic neurosurgery,* Image and Vision Computing Volume, pp. 468-485, 1993

[42] X.S Chen, C.K. Chui, S.H.Teoh, S.H. Ong, W.L. Nowinski, *Automatic modeling of anatomical structures for biomechanical analysis and visualization in a virtual spine workstation*, the 4th International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI), 2001.

[43] A. Pommert, *Volume Visualization in Medicine: Techniques and Applications, Focus on Scientific Visualization*, Springer Verlag, pp. 41-72, 1993

[44] C. Pisupati et. Al, *A Central Axis Algorithm for 3D Bronchial Tree Structures*, Proc. IEEE International Symposium on Computer Vision, pp. 76-82, Nov. 1995

[45] S. Wood, E. Zerhouni, J. Hoford, and E. Hoffman, *Measurement of Three-dimensional Lung Tree Structures by Using Computed Tomography*, Journal of Applied Physiology, 79:1687-1679, 1995

[46] G. Bertrand and G. Maladain, *A new topological classification of points in 3D images,* In European Conference on Computer Vision, pp 169-175, 1992

[47] P. Dokládal, C. Lohou, L. Perroton, and G. Bertrand, *A new thinning algorithm and its application to extraction of blood vessels*. In Biomedisim'99, pp.32-37. ESIEE, 1999

[48] K. Palágyi, E. Sorantin, E. Balogh, A. Kuba, Cs. Halmai, B. Erdôhelyi, K. Hausegger, *A sequential 3D thinning algorithm and its medical applications*, Lecture Notes in Computer Science 2082, pp 409-415, Springer, 2001

[49] I. Nyström, Ö. Smedby, *Skeletonization of volumetric vascular images - Distance information utilized for visualization,* Journal of Combinatorial Optimization, 5(1): 27-41, 2001

[50] P.P. Anna, *Discrete Medial Axis Transform for Discrete Objects*, Technical Report LSI-98-22-R, Universitat Politècnica de Catalunya, 1998

[51] M. Näf and O. Kübler and R. Kikinis and M.E. Shenton and G. Székely, *Characterization and Recognition of 3D Organ Shape in MedicalImage Analysis Using Skeletonization*, In Workshop on Mathematical Methods in Biomedical Image Analysis, pp 139-150. IEEE, 1996.

[52] R.L. Ogniewicz and M. Ilg, *Voronoi skeletons: Theory and applications*, In IEEE Conf. of Computer Vision and Pattern Recognition, pp 63-69, 1992.

[53] D. Chen, B. Li, Z. Liang, M. Wan, A. Kaufman, and M. Wax, *A tree-branch searching, multi-resolution approach to skeletonization for virtual endoscopy,* In K. Hanson, editor, Medical Imaing – Image Processing, volume 3979 of Proceedings of SPIE, pp.726-734, 2000.

[54] S. Flesia and M. Roche, *3D reconstruction of generalized cylinders from biplane projection,* Computer and Graphics, 16:167-173, 1992.

[55] M.S. Kim, E.J Park, and H.Y Lee, *Modelling and animation of generalized cylinders with variable radius offset space curves,* The Journal of Visualization and Computer Animation, Vol.5, No. 4, pp.189-207, 1994

[56] R. Barzel, *Physically-based Modelling for Computer Graphics: A Structured Approach,* Academic Press, INC., London, 1992

[57] W. Hua, C.K. Chui, Y.P. Wang, Z.L. Wang, X.S. Chen, Q.S. Peng and W.L. Nowinski, *A Semiautomatic Framework for Vasculature Extraction from Volume Image,* The 10th International Conference on Biomedical Engineering, pp. 517-518. 2000.

[58] X.S. Chen, *Vascular extraction and modeling for medical simulation and visualization*, pp.13-14, Master thesis, NUS, 2002

[59] D.R. Yi and V. Hayward, *Skeletonization of volumetric angiograms for display,* Comput Methods Biomech Biomed Engin, Vol. 5, No. 5, pp. 329-341, 2002

[60] C. Loop, *Smooth subdivision surfaces based on triangles,* Master's thesis, University of Utah, Department of Mathematics, 1987

[61] E. CatMull, and J. Clark, *Recursively generated B-spline surfaces on arbitrary topological meshes,* Computer Aided Design 10(6): 350-355, 1978

[62] A. Gregory, M. Li, S. Gottschalk, and R. Taylor, *H-COLLIDE: A Framework for Fast and Accurate Collision Detection for Haptic Interaction,* IEEE Virtual Reality Conference, 1999

[63] S. Gottschalk, M. Lin, and D. Manocha, *OBB-Tree: A Hierarchical Structure for Rapid Interference Detection,* In Proc. of ACM Siggraph'96, pp.171-180, 1996

[64] X. Ma, Y. Lu, L.K. Fock,W.L. Nowinski, *Haptic device for web-based training system for interventional radiology procedures,* Computer Assisted Radiology and Surgery 2003

[65] X. Ma, H. Marcelo, J. Ang., and C.K. Chui, *Portable Haptic Interaction Device Using Micro-catheter for IR Simulation*, Preprints of 2nd IFAC Conference on Mechatronic Systems, pp. 317-322, Berkeley, California, USA, 2002

[66] Y. Lu, X. Ma, K.F. Loe, C.K. Chui, and W.L. Nowinski, *Haptic Vascular Modeling and Visualization in Web-enabled Interventional Neuroradiology Simulation System,* Accepted in Medical Imaging 2004