THE THESIS OF DOCTOR OF PHILOSOPHY

Study on a Virtual Reality-based Training System for the Vascular Interventional Surgery

Jin Guo

Graduate School of Engineering
Kagawa University

Abstract

Cardiovascular and cerebrovascular diseases are the leading causes of deaths worldwide. Based on a World Health Organization survey, it is estimated that, till 2020, coronary artery disease and cerebral hemorrhage are still the first and second causes of death of human being. Catheter-based endovascular interventional procedures have been extensively adopted for treatment and diagnosis of various vascular diseases. During the procedures, a flexible catheter having a pre-bent shape with natural curvature at the tip is directed inside the patient's vascular structure, from the femoral artery to the target position, to cure the vascular diseases by manipulating the tail part of the catheter. The gradually increasing success rate of catheter-based interventions, combined with their less blood loss and minimal damage to healthy tissue, has lead to a significant growth in the number of procedures performed annually. However, it is still a challenging work to position a catheter in a target vessel branch within the highly complicated and delicate vascular structure. The interventionalists have to carry out the operations by imagining the spatial relationship between the catheter and its surrounding vessels based on their experience and feelings of hands due to the lack of intuitive visual feedback. Additionally, complicated vessel shapes and multi contacts between the catheter and the blood vessel walls lead to significant deterioration in the feasibility and maneuverability of positioning the

catheter. The catheter manipulation skills of experienced interventionalists are therefore critical to the success of the endovascular catheterization.

Compared to conventional training methods on cadavers, animals, phantoms or real patients, the VR-based simulators provide not only the flexible and repeatable scenarios but also the possibility to precisely assess training progress. The virtual reality-based simulators for training in Vascular Interventional Surgery (VIS) were presented in this paper. This application allows generating realistic physical-based models of catheters and blood vessels, and enables surgeons to touch, feel and manipulate virtual catheter inside a vascular model through the same surgical operation mode as is used in actual VIS. Experimental results show that the error rate is in an acceptable range and that simulators can be used for surgery training.

Additionally, most of the existing controllers for virtual reality-based simulators introduced commercially available interfaces (e.g. PHANTOM Omni) or a remote controller for game machines (e.g. Nintendo Wii) or robotic control consoles which are manipulated, unfortunately, not in a manner that is in accordance with conventional catheter operation practices. Furthermore, few researchers focused on the controller designs with much consideration of the natural catheter manipulation skills obtained from manual catheterization. A new master controller for endovascular catheterization that can allow the

interventionalists to apply conventional pull, push and twist of the catheter as used in current practice was presented in this paper. A catheter sensing unit (used to measure the motion of the catheter) and a force feedback unit (used to provide a sense of resistance force) were both presented. A camera was used to allow a contactless measurement that avoids additional friction and the force feedback in the axial direction was provided by the magnetic force generated between the permanent magnets and the powered coil. Performance evaluation of the controller was evaluated by first conducting comparison experiments to quantify the accuracy of the catheter sensing unit, and then conducting several experiments to evaluate the force feedback unit. From the experimental results, the minimum and the maximum errors of translational displacement were 0.003 mm and 0.425 mm respectively. The average error was 0.113mm. In terms of rotational angles, the minimum and the maximum error were 0.39 ° and 7.2 ° respectively. The average error was 3.61 °. The force resolution was approximately 25mN and a maximum current of 3A generated an approximately 1.5 N force. Based on analysis of requirements and state of the art in computer-assisted and robot-assisted training systems for endovascular catheterization, a new master controller with force feedback interface was proposed to maintain the natural endovascular catheterization skills of the interventionalists.

Most importantly, to realize the remote training course, the virtual reality-based method for reducing visual feedback transmission times based on an Internet-based tele-operative robotic catheter operating system was designed for vascular interventional surgery, to reduce radiation-exposure times, and afford unskilled surgeons the opportunity to learn basic catheter/guidewire skills, while allowing experienced physicians to perform surgeries cooperatively. Remote surgical procedures, limited by variable transmission times for visual feedback, have been associated with deterioration in operability and vascular wall damage during surgery. At the patient's location, the catheter shape/position was detected in real time and converted into three-dimensional coordinates in a world coordinate system. At the operation location, the catheter shape was reconstructed in a virtual-reality environment, based on the coordinates received. The data volume reduction significantly reduced visual feedback transmission times; coordinate values were easily synchronized with control data and haptic signals. Remote transmission experiments, conducted over inter-country distances, demonstrated the improved performance of the proposed prototype.

Contents

List of Figures

Acknowledgements

The author wishes to express his great gratitude to his supervisor Professor Shuxiang Guo for his invaluable guidance, support and friendly encouragement throughout my Ph.D. course and for providing me with first class resources.

The author would like to thank Prof. Hideyuki Hirata and Prof. Keisuke Suzuki. Thanks them for their valuable advices and suggestion on my research. Without their help, I can hardly finish this thesis.

Also the author would like to acknowledge the efforts of his laboratory members, especially Dr. Nan Xiao and Ms. Maoxun Li whose time and expertise were greatly appreciated.

This research was supported by Kagawa University Specially Promoted Research fund 2012.

Finally, the author gives special appreciation to his parents for their love, patience and support.

Declaration

I hereby declare that this submission is my own work and that to the best of my knowledge and belief. It contains no material previously published or written by another person nor material which to a substantial extent has been accepted for the award of any other degree or diploma of the university or other institute of higher learning, except where due acknowledgment has been made in the text.

Chapter 1

Introduction

1.1 Preface

In this thesis, the virtual reality-based simulators for surgeons' training in vascular interventional surgery were proposed and developed. These simulators were capable of generating the realistic virtual-reality environment of blood vessels according to a patient's special computed tomography (CT) or magnetic resonance imaging (MRI). In addition, the new master controller had the ability to allow the surgeons to directly employ a real catheter as the controller maintaining the surgeons' conventional operating skills in an actual surgery. It can also allow the surgeons to carry out interventions with force feedback in the axial direction, which provides the surgeon with a sense of resistance.

(1) To reconstruct a realistic three-dimensional vascular model, a median filter algorithm was employed to reduce the noise of CT images. Next was to segment the blood vessel walls from the CT or MRI images based on the level set algorithm. Then the 3D surface of a vascular segment from CT or MR images was rebuilt by using marching cube algorithm. Finally, centerlines of a vascular segment

were computed and the surface of a vascular segment was smoothed.

(2) To maintain the natural endovascular catheterization skills of the interventional radiologist, a camera was employed to detect the axial and radial motions of a catheter simultaneously with the ability to allow interventional radiologists to use their dexterous skills while performing catheter-based interventions. Additionally, a force feedback unit was developed to provide the interventionalists with a sense of resistance force.

(3) To realize the distance teaching and learning, the virtual reality-based method was developed to effectively reduce the transmission time of visual feedback due to the significant reduction in data volume using coordinates instead of actual images.

1.2 Background

With the quickening pace of modern life, the brain diseases of people are increasing, such as cerebral aneurysm and infarction and so on. The traditional surgery spends patients a lot of operation time and has long recovery time; the burden on patients is heavy. Vascular Interventional Surgery (VIS, Shown in Figure 1-1) is a revolutionary surgical technique, in which surgery is done using precise medical devices and viewing equipment inserted through a small incision instead of making a large incision to expose the operation site. A lot of diagnosis and medical surgery with an endoscope or a catheter are performed for vascular interventional surgery recently. The procedure of catheter ablation is shown in Figure 1-2 and the Figure 1-3 presents the coil embolization surgery. The diagram of the stent delivery is shown in Figure 1-4, and the Figure 1-5 shows the drug delivery surgery. The main advantage of this technique is to reduce trauma to healthy tissue since this trauma is the leading cause of patient' pain and scarring and a prolonged hospital stay. Fast recovery and a short hospital stay also reduce the cost of surgery and radiation exposure. A critical disadvantage of this surgery technique, however, is its complexity, requiring extensive training efforts of the interventionalists to achieve competency because the arteries through which the catheter passes are extremely intricate and delicate. It is difficult for novice surgeons to insert the catheter from femoral to the target position within the patient's vascular structure especially the blood vessel of patient's brain. In addition, repeated insertion of the catheter through several tests could tear a blood vessel and cause bleeding instantaneously, while excessive pressure could rupture blood vessels. For practical and ethical reasons, realistic virtual-reality simulators provide a promising alternative method compared to other available alternatives such as anesthetized animals, human cadavers and patients. VR-based simulators enable novice interventionalists to learn basic wire and catheter handling skills and provide expert practitioners with the opportunities to rehearse new operation procedures prior to performing them on a patient.

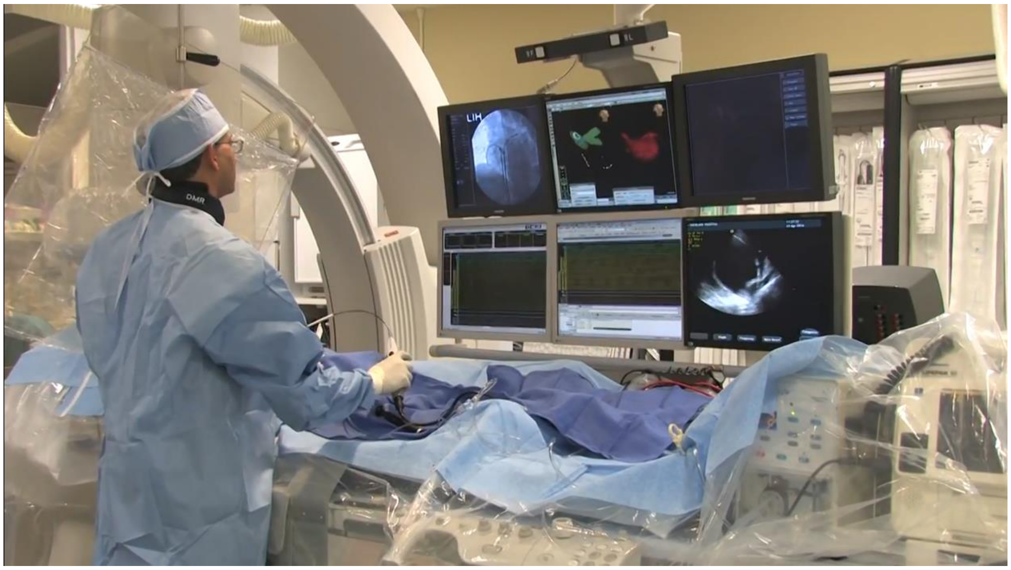


Figure1- 1 Vascular Interventional Surgery (VIS)

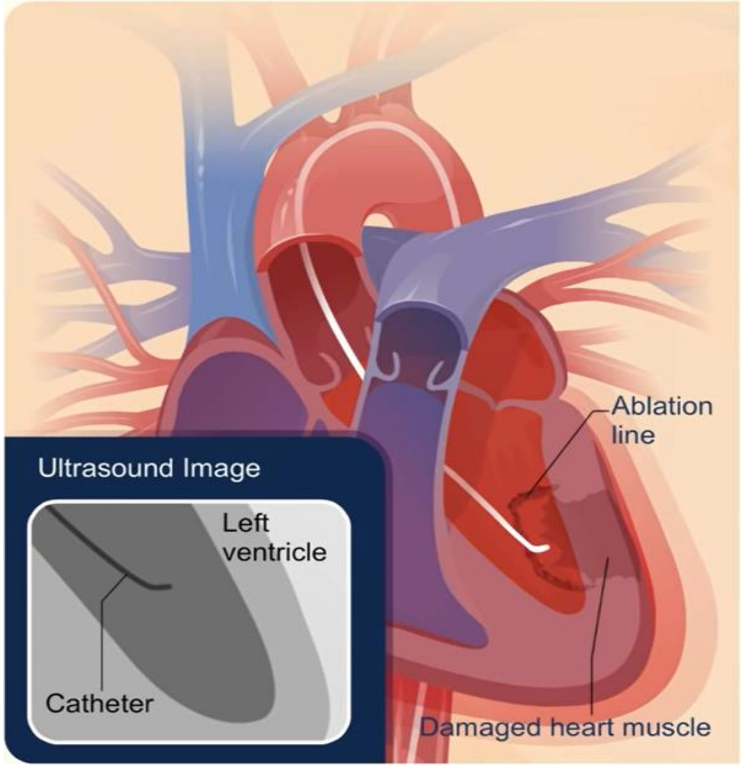


Figure1- 2 the procedure of catheter ablation [Padalino13]

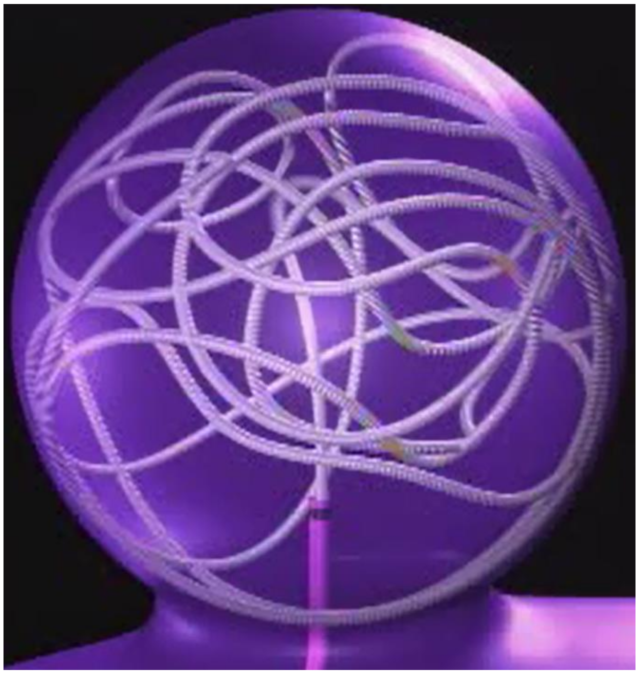


Figure1- 3 the procedure of aneurysm coil embolization [Spiotta12]

Additionally, distance teaching/training has become indispensable to novice surgeons around the world. As an example, the use of this advanced minimally invasive technique in Canada has been severely hampered by the inability to provide adequate training and support to community surgeons, many of who live in remote regions of the country [Anvari07]. If a remote surgical system is in place, a competent surgeon can work cooperatively to assist in a surgical procedure that takes place in another region [Mitsuishi07]. However, due to long transmission times and the variable quality of the visual image feedback, these surgeons may be exposed to prolonged periods of radiation [Guo12]. A more efficient, safe, and reliable method is needed to disseminate new surgical knowledge worldwide.

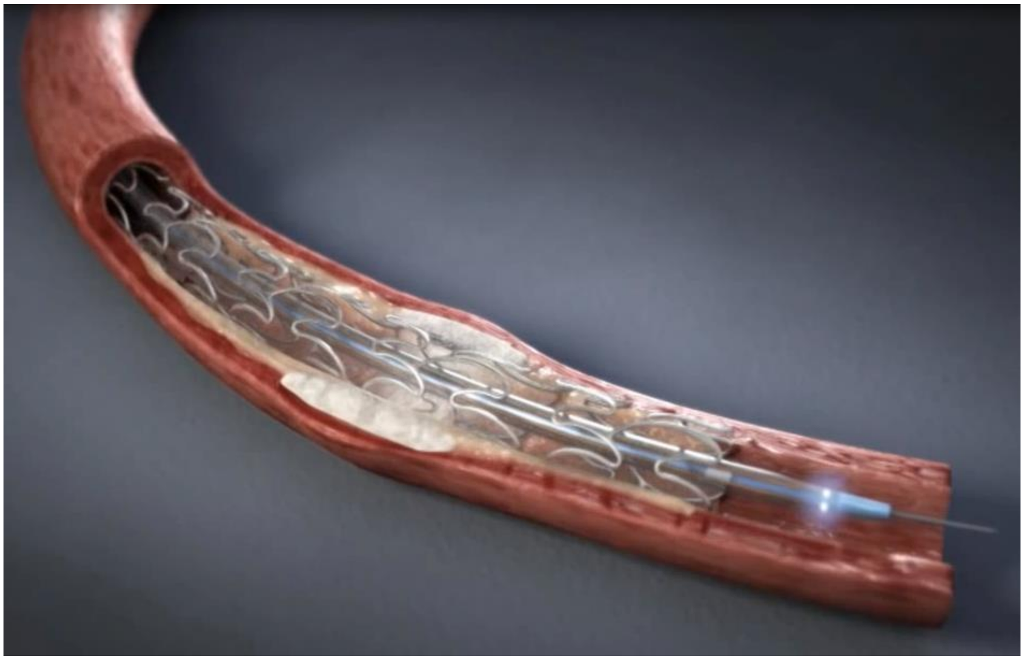


Figure1- 4 the procedure of stent delivery

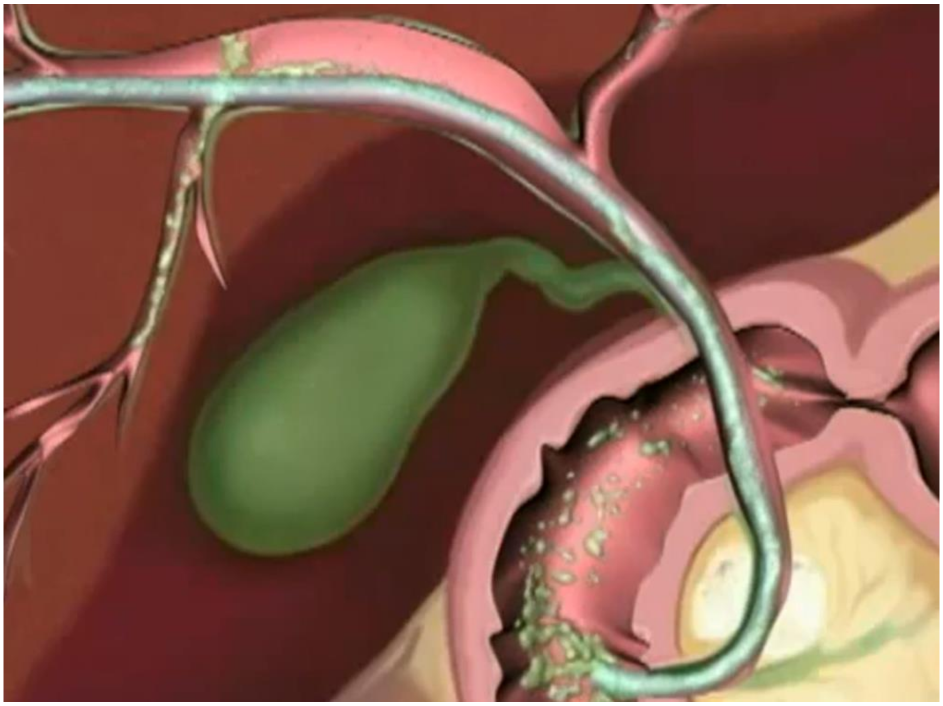


Figure1- 5 the procedure of drug delivery

Based on aforementioned background, in this paper, a virtual reality-based catheterization training system for vascular interventional surgery was proposed and developed. The simulators were capable of extracting blood vessel walls from the specific patients' CT or MRI images. Physics-based deformation model of the catheter and the blood vessel walls were designed and developed in the virtual reality-based training system. A new master controller with force feedback was designed and developed to maintain the natural endovascular catheterization skills of the interventional radiologist. For distance training and teaching, a newly developed method was described, based on virtual reality and image-processing technology.

1.3 Literature review

Compared to conventional surgical methods, vascular interventional surgery has many advantages. Therefore, Catheter-based endovascular interventional procedures have been extensively adopted for treatment and diagnosis of various vascular diseases. During the procedures, interventionalists manipulate the tail part of the catheter to target branch vessels to cure the vascular diseases [Guiatni13]. Thus complicated vessel shapes and multi contacts between the surgical instrument and the vessel walls inevitably lead to significant deterioration in the feasibility and operability of positioning the catheter [Srimathveeravalli10].

To afford novice interventionalists the opportunity to learn basic catheter manipulation skills, robot-assisted and computer-assisted catheterization training is two promising methods. Two commercially available endovascular simulators are the Procedicus Vascular Interventional System Trainer (shown in Figure 1-9) [VIST, MenticeAB][Aggarwal06] (VIST, Mentice AB, Gothenburg, Sweden) and the Angio Mentor^{TM} (shown in Figure 1-8) [Simbionix, Ltd.][Benjamin14] (Simbionix, Ltd., Cleveland, OH, U.S.), which allows novice interventionalists to conduct the comprehensive and goal-oriented training and education program before treating real patients including catheter/wire manipulating, stent deployment and angioplasty balloon inflation in the domain of renal, carotid, aortoiliac and peripheral vascular structures [Desser07]. The Sensei robotic navigation system (shown in Figure 1-6) [HansenMedical][Saliba08] (Hansen Medical, Mountain View, Calif) and the Niobe magnetic navigation system (shown in Figure 1-7) [Stereotaxis][Chun08] (Stereotaxis, St. Louis, Mo) are two typically commercial surgical robotic systems that have been used in the endovascular interventional surgery with electromechanical base and magnetically controlled mechanisms of action. They have been successfully used in different clinical applications including cardiac ablation and endovascular aneurysm repair [Payne12].

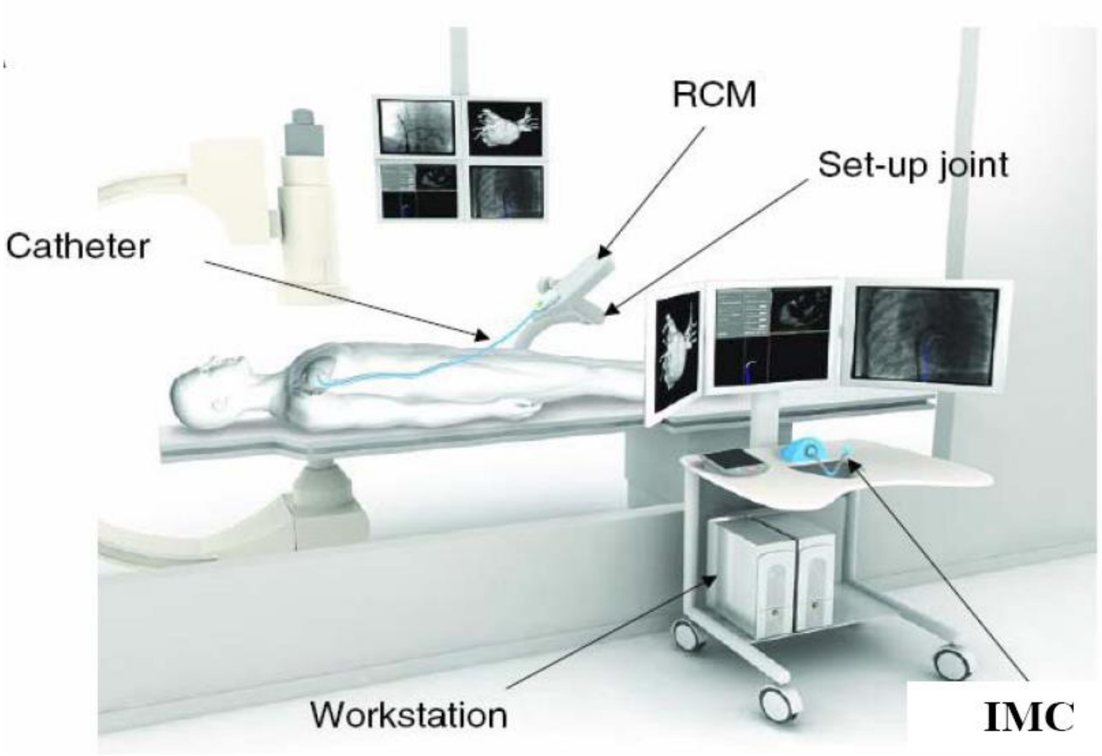


Figure1- 6 the Sensei Robotic Catheter System [HansenMedical]

However, there are several limitations to the application of these commercial products. The most obvious one is the expensive cost of the systems, which makes them cost prohibitive to most private centers [Antoniou11, Das14, Binning12]. Additionally, the catheters must be integrated with their custom interfaces and navigation systems to do the catheterization operations, which cannot be widely available.

In addition to the commercially available products, many research groups have studied endovascular interventional training systems. Cercenelli et al. [Cercenelli07][Marcelli08] developed a novel tele-robotic system to manipulate standard electrophysiology catheters. However, there is no force feedback device in their controller design.

The loss of force feedback in robotic catheter systems can increase the risk of damage to the blood vessel walls. Fukuda et al. [Tanimoto97, Negoro02, Tercero12] at Nagoya University have developed a Joy Stick as the controller of the master system with a haptic unit to handle the manipulators of the slave system. There are two joints in the design of the master controller and each joint was connected to a motor with a wire mechanism. Another research group in [Guo12] designed a master controller to simulate the conventional push, pull and rotate motion used in current practice. However, unlike the conventional endovascular catheterization, in which interventionalists manipulate an actual catheter directly using their hands, the employment of these master controllers replaces the catheter from the interventionalists' hand with the "handle part", which are made of metal materials with different size and material attributes from a real catheter, to carry out catheterization training operations. The loss of the hand feelings on a real catheter unavoidably contaminates their intuitive and dexterous skills in the interventional radiology procedure. To achieve better training effects, Yogesh Thakui et al. [Thakui09] developed a novel remote catheter navigation system to directly manipulate percutaneous transluminal catheters from a location far away from the patient while allowing the interventionalists to apply conventional insertion, extraction and rotation motion in the procedure. A similar design was presented by the researchers in [Ma13]. However, these research achievements have a common defect: To measure the motion of the input catheter, the catheter is needed to be clamped by the rollers coupled with encoders. To ensure the catheter does not slip in the apparatus, a force has to be applied to the rollers, which leads to the difficulties in designing the force feedback device to provide the interventionalists with a sense of resistance force.

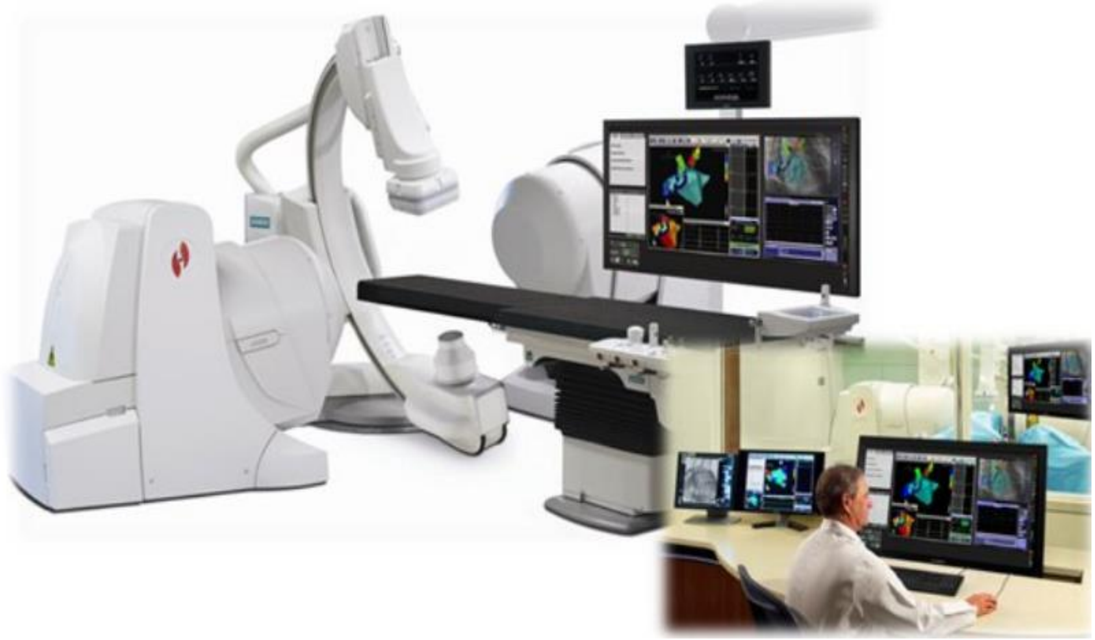


Figure1- 7 the Stereotaxis Niobe II [Stereotaxis]

Because vascular interventional techniques have unavoidably reduced the sense of touch compared to open surgery, surgeons have to rely more on the haptic feeling generated by the interaction between blood vessels and the catheter. Even if the color and texture of blood vessels visually convey crucial anatomical information, touch is still critical in surgery. The benefits of using force feedback devices in vascular interventional surgery training through simulation have already been recognized by several research groups and many of the companies working in this area [Basdogan01, Çakmak00, Brown02, Tsang08, Wei09, Basdogan07, Chiang11, Takashima07, Meier01, Aggarwal06, Neequaye07]. In these researches, however, the virtual surgical trainings were carried out without haptic feedback, or researched on a virtual model of body organs not a vascular physical model. Moreover, some achievements in this area have used commercial products or other haptic devices as a controller to in virtual vascular interventional surgery [Basdogan04]. Nevertheless, it is not convenient when surgeons drive the catheter for insertion and rotation because it does not accord with the custom of surgeons' operations.

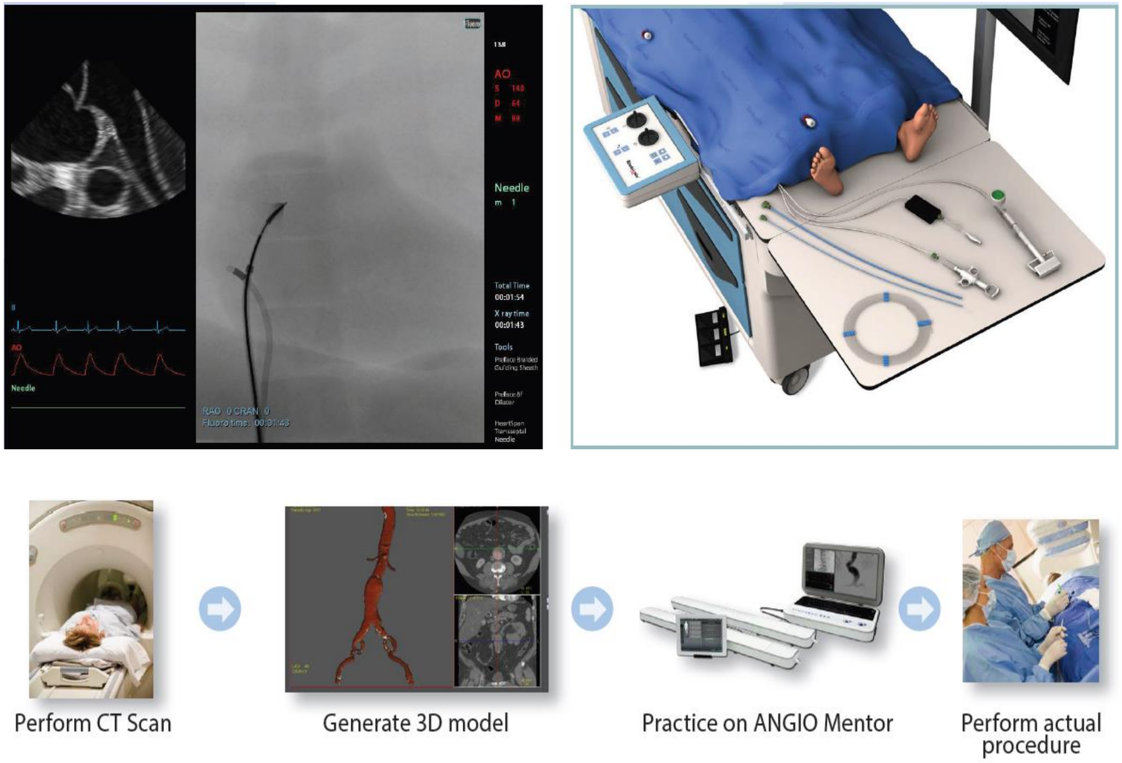


Figure1- 8 the Angio Mentor™ System [Simbionix, Ltd.]

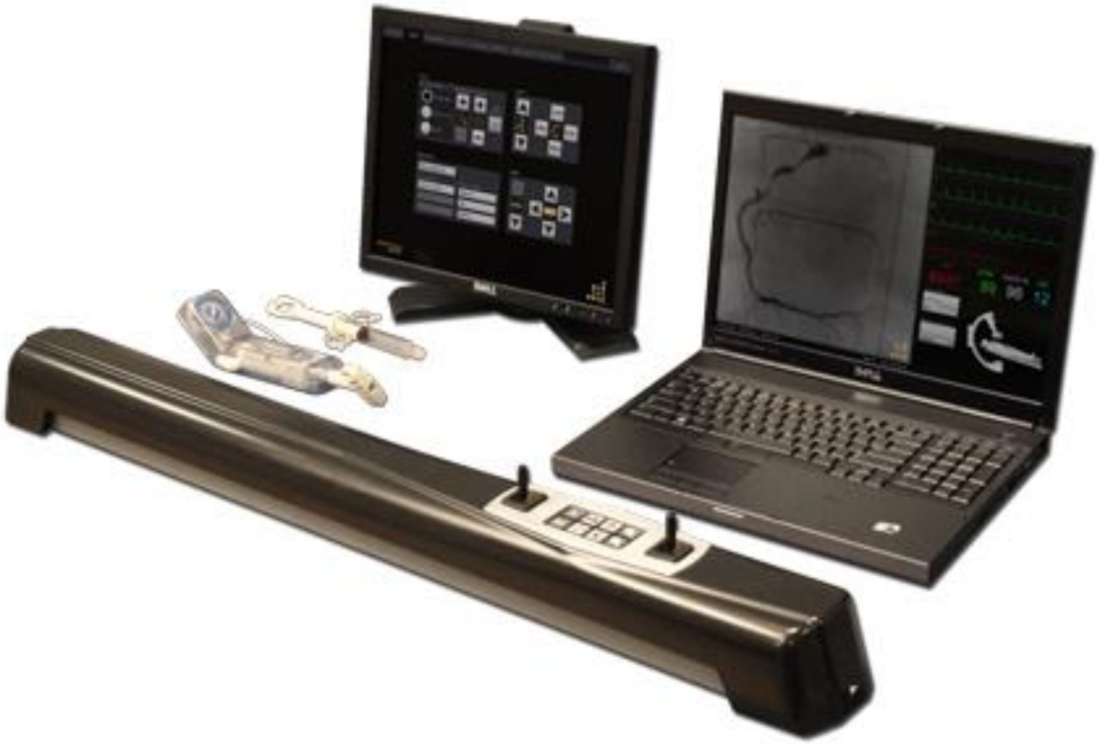


Figure1- 9 the Procedicus Vascular Interventional System Trainer [VIST, MenticeAB]

Due to practical and ethical reasons, computer-assisted training systems have distinct advantages over traditional training methods on cadavers, animals, phantoms or real patients. The VR-based simulators are capable of providing not only the more realistic scenarios but also the flexible training courses. However, most of the existing controllers introduced commercially available interfaces [Srimathveeravalli10] (e.g. PHANTOM Omni) or a remote controller for game machines [Wei09] (e.g. Nintendo Wii) or robotic control consoles [Lu11] which are manipulated, unfortunately, not in a manner that is in accordance with conventional catheter operation practices.

To address the need for providing a familiar and ergonomic setting for the interventionalists, a compact master controller capable of measuring the radial and axial motion of the input catheter simultaneously and providing force feedback while still preserving the manual catheter manipulation skills was designed. A camera was used to allow a contactless measurement that avoids additional friction and the force feedback in the axial direction was provided by the magnetic force generated between the permanent magnets and the powered coil.

Because the vascular structure through which the catheter passes is highly intricate and delicate, VIS requires extensive training by the surgeon to achieve competency. Thus, distance teaching/training has become indispensable to novice surgeons around the world. To meet these requirements, the Internet-based remote catheter operating system provides a promising method to realize participation of highly capable physicians in surgical procedures that take place in distant locals [Liu11, Arata07, Arata06]. In 2001, for example, researchers in [Marescaux01] performed remote surgery using a surgical robot, ZEUS, via a dedicated asynchronous transfer mode (ATM) between New York and Strasbourg, France. However, a dedicated fiber-optic network for remote surgery would be costly. Conventional existing network infrastructures, such as the Internet, provide a preferable conduit for future remote surgical procedures [Arata07].

Many research groups have studied robotic telesurgical systems based on the Internet [Zhou07, Al-Mouhamed05, Islam11, Xiao12, Kim10, Xie07, Smith10]. For example, the Centre for Minimal Access Surgery (CMAS) at McMaster University used broadband Internet and telecommunication systems to provide distance training and mentoring to community surgeons living in remote northern communities of Canada [Anvari07]. Generally, the concept of an Internet-based surgical system consists of three parts: a master system at the operation location, a slave system at the patient's location, and a communication link, as shown in Figure 7-1. The master and slave systems are connected by transmission control protocol/Internet protocol (TCP/IP) networks for sending and receiving data, including image information, haptic feedback, and manipulator control signals.

The time delay for visual transmission is unavoidable with Internet-based remote surgical systems. The visual information is much larger than the control signals and haptic feedback. Thus, the incremental time delay for visual information is always associated with significant deterioration in the feasibility and maneuverability of surgical tools. This delay thus decreases the effectiveness of the training session during remote surgical education, and has the potential to contribute to surgical damage to the surrounding tissue during the procedure. To date, researchers have focused mainly on delayed control and force signals via the Internet [Natarajan08, Richert09]; however, few studies have examined the time delay associated with visual transmission. Conventionally, delays in visual information retrieval have been handled by predictive methods, increased autonomy [Smith10], or high-speed image compression processes [Jayender08]. However, most prediction-based delay compensation methods rely on predicting the feedback from the remote environment, which may be difficult to model, particularly if the main feedback is a video signal. Additionally, the image compression process is time-consuming.

To solve these problems, a newly developed method was proposed, based on virtual reality and image-processing technology, for efficiently reducing visual feedback transmission times, making it easier to control the time difference between haptic signals, control data, and image information. This was achieved by reducing the amount of communication data in the image information. Several remote visual feedback transmission experiments were conducted to demonstrate the feasibility of the proposed prototype, via a cooperative surgical effort between research groups in China and Japan. The tele-operative catheter operating system proposed in this work facilitates teamwork between medical professionals worldwide; additionally, this technique minimizes radiation exposure times by reducing the transmission time for visual feedback.

1.4 Contributions

In this thesis, a virtual reality-based endovascular catheterization training system was designed and developed. Firstly, a hybrid 3D reconstruction method was proposed to extract the blood vessel walls from the specific patients' CT or MRI files. Physics-based deformation models were built to simulate the behavior after the interaction between the blood vessel walls and the virtual catheter. A new master controller with force feedback for the virtual reality training system was designed and developed to allow the interventionalists to do the surgical training courses by using the natural endovascular catheterization skills. To solve the high cost and difficulties in distance teaching in Canada, a virtual reality and image-processing based method was described to improve the feasibility and maneuverability of the remote learning and traing course.

The mainly contributions of the virtual reality-based systems are as following:

(1) Flexible reconstruction methods for realistic three-dimensional vascular models based on the specific patients' CT or MR images. Core components of virtual-reality surgical simulators and training systems are the realistic vascular models that are virtual representations of real blood vessels that display accurate displacement due to the fact that models with incorrect material properties and shapes could result in

adverse training effects. In this study, two kinds of reconstruction algorithms were proposed. One is to rebuild the vascular structure based on normal and advanced CT or MRI files. The normal images were employed to subtract their corresponding advanced images to remove human bones and other soft tissues, thereby remaining the data of blood vessels. This approach is adaptable to calculate the whole vascular structure. The other one is employed to reconstruct the 3D vascular models based on advanced CT or MRI files only according to marching cube and level sets algorithms. This approach is preferable to build a vascular segment due to its flexibility.

(2) Development of a new master controller for the virtual reality-based training system. Based on the requirements of the interventionalists and the state of the art in the design of the master controller for computer-aided and robot-aided training systems for endovascular surgery, a new master controller capable of contactless measuring the axial push-pull and twisting motions of a catheter with force feedback to the operators was proposed, whilst still preserving the natural ergonomic skills of conventional catheterization. A camera was employed to detect the motions of the catheter freely and interventionalists can manipulate the real catheter intuitively with a familiar, ergonomic setting. Force feedback to the interventionalists in axial direction was generated by Magnetic force based on the calculated forces between the vascular model and the virtual catheter in a virtual

reality environment or the forces from the sensing unit of the slave system. Several experiments were conducted to evaluate the performance of the new master controller and the experimental results showed that the errors of the catheter sensing unit and haptic unit of the new master controller were in an acceptable range and it can be used as a controller in a computer-aided or robot-aided training system.

(3) Design of a new prototype to reduce the transmission time of visual feedback based on virtual reality and image processing algoritms. In this part, a new prototype based on the Internet was proposed and developed, for efficiently reducing the transmission time of the visual feedback. It also can make it easier to control the time difference between haptic signals, control data, and image information by reducing the amount of communication data in the image information. Several remote visual feedback transmission experiments were conducted to demonstrate the feasibility of the proposed prototype, via a cooperative surgical effort between research groups in China and Japan. The purpose of the experiments was to evaluate the performance of the new catheter operating system and its associated transmission times for visual feedback, by comparing performance before and after the application of the proposed architecture. The maximum error for catheter reconstruction on the tip of the catheter was 0.93 mm less than 1 mm in these 10 frames. Remote transmission experiments showed that the transmission time of the visual feedback was reduced

significantly based on the newly developed method.

1.5 Thesis structure

Figure 1-10 shows the structure of the thesis. Chapter 1 is the introduction. It is composed of preface, background, literature review, research purposes, research approaches and thesis structure. In chapter 2, overview of the previous and proposed virtual reality-based training systems are compared and introduced. Chapter 3 introduces the 3D vascular reconstruction algorithms. Two reconstruction methods were proposed and developed. The specific patients' CT or MRI images were the input of the reconstruction algorithms. Level sets and marching cubes algorithms were used to segment the blood vessel walls and to generate the surface model of the blood vessels. In chapter 4, physics-based models for the virtual reality-based training system has been described. The mass-spring model was employed to simulate the deformation behavior of the blood vessel walls to obtain better performance. The elastic-joint-based method to simulate the deformation behavior of the virtual catheter model has been introduced. In chapter 5, a new haptic controller for training in vascular interventional surgery has been developed. Several experiments were conducted to evaluate the performance of the new master controller. Chapter 6 presented the 2D and 3D models for reducing the time delay of visual feedback based on image processing and virtual reality techniques. In chapter 7, conclusions and future work were given.

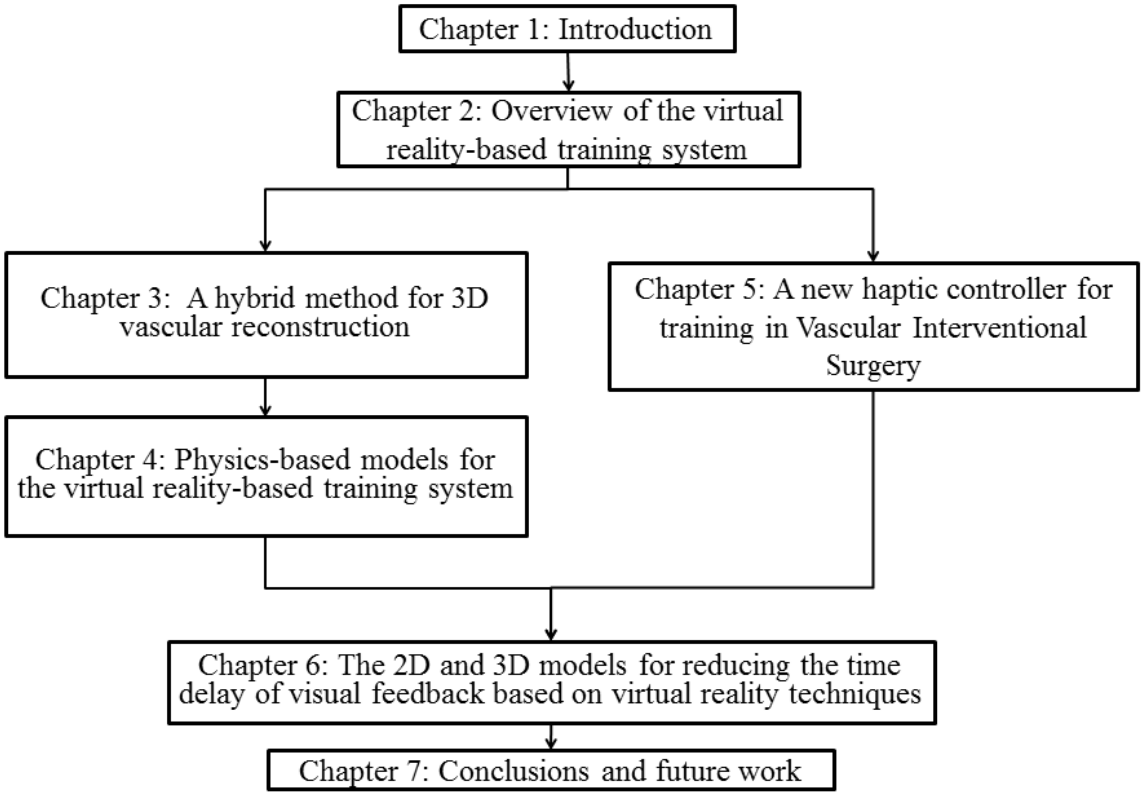

Chapter 1: Introduction
Chapter 2: Overview of the virtual reality-based training system
Chapter 3: A hybrid method for 3D vascular reconstruction
Chapter 5: A new haptic controller for training in Vascular Interventional Surgery
Chapter 4: Physics-based models for the virtual reality-based training system
Chapter 6: The 2D and 3D models for reducing the time delay of visual feedback based on virtual reality techniques
Chapter 7: Conclusions and future work

Figure1- 10 The structure of the thesis

Chapter 2

Overview of the virtual reality-based training system

There are several conventional methods for the training of catheterization skills such as animals, anatomic phantoms and actual patients. However, animals have the distinct anatomical differences with human being and it is very difficult and expensive to produce anatomic phantoms due to the lack of flexibility. Obviously, acting directly on the actual patients is very dangerous to both patients and trainees. Compared to conventional training methods, virtual reality-based catheterization training systems have distinct advantages such as high flexibility, high realism, low cost and without any risks. It allows not only training novice surgeons with basic catheter manipulation skills and enabling them to learn new catheterization courses but also designing a surgical pre-planning treatment in the VR system to better accomplish the operation before an actual surgery is performed.

2.1 The previous virtual reality-based simulators

Our laboratory has been working on the virtual reality-based simulators for endovascular catheterization training. The first VR-based training system is shown in Figure 2-1.

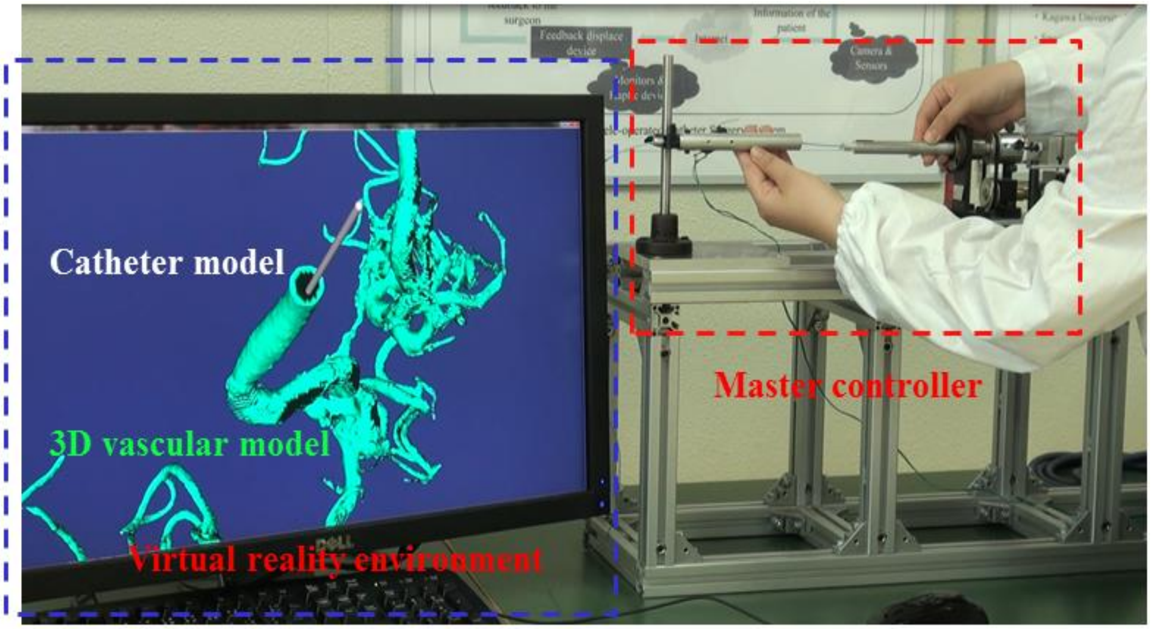


Figure 2- 1 The first virtual-reality-based training system

On the master side, a surgeon operated a handle to drive a catheter for insertion or rotation, which, to the surgeon, seems to clamp catheter directly. At the same time, control instructions of the catheter operating system were transmitted to the virtual-reality environment. The catheter in the virtual-reality environment was inserted or rotated according to control commands from the master side. However, there are some limitations for this VR-based training system. It is lack of deformation for catheter and blood vessel walls and it is absence of the collision detection between the catheter model and the 3D vascular model in the vritual reality environment. In terms of the master controller, it is incapable of using an actual catheter to do the training courses.

Thus, the deformation model of the blood vessel walls and the catheter and the collision detection algorithms were added to the virtual reality training system. The improved VR-based training system in vascular interventional surgery was shown in Figure 2-2. If the catheter contacts a blood vessel, the collision vectors between the simulated catheter and the blood vessel walls was detected, stored and the deformation steps were carried out to provide the better performance of visual feedback to the novice surgeons. Based on collision vectors and three-dimensional vascular image information, the virtual-reality environment can be used for medical training.

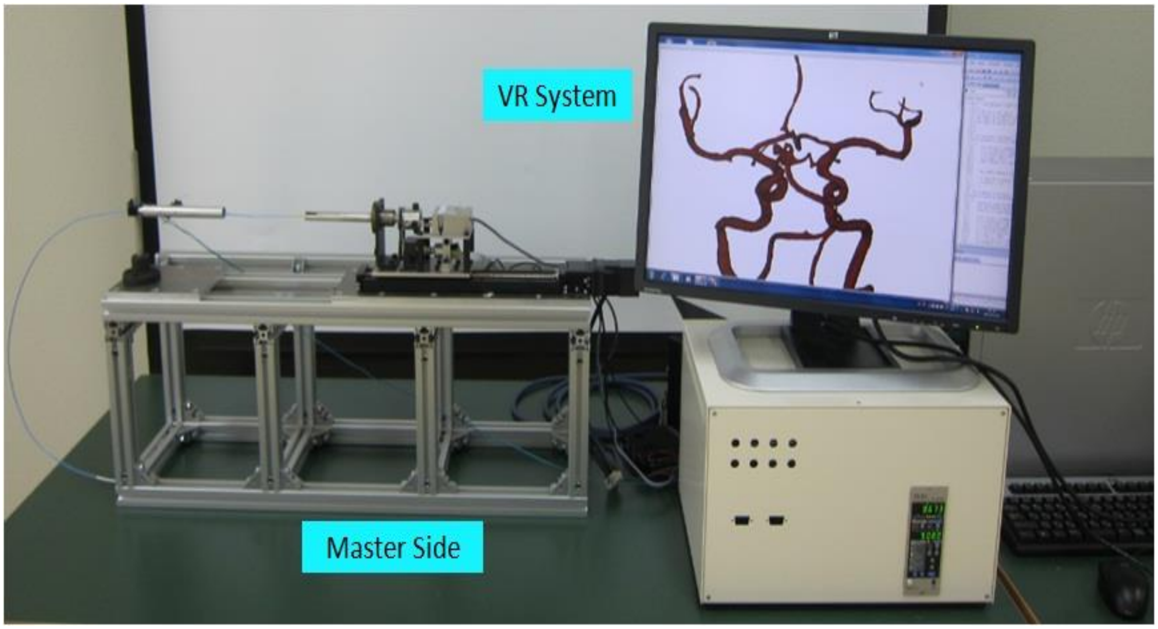


Figure 2- 2 The improved virtual-reality-based training system

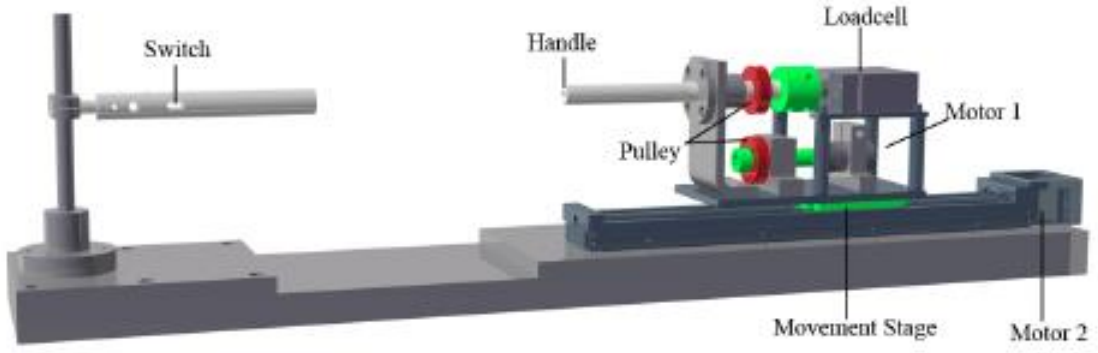


Figure 2- 3 The design of the master controller for the VR-based training system.

The surgeon’s console was the master side of the entire system, as shown in Figure 2-3. Surgeons carried out operations using the console. The switch on the left handle was used to control the two graspers at the slave side; only one switch was necessary, because the catheter was clamped by only one grasper at a time. The operator’s action was detected using the right handle. The movement part of the catheter manipulator mimicked the same motion as the right handle of the controller. The right handle measured both the axial and radial movement of the surgeon’s hand. The handle was sustained by a bearing, linked to a load cell; a pulley was attached to the handle. A dc motor (Motor 1) with an encoder generated torque feedback. A pulley which is couple to the upper one is fixed to the axle of the motor. All of the parts mentioned above were placed on a translation stage, which was driven by a stepping motor (Motor 2). Axial movement was measured as follows. The pulling/pushing force was measured by the load cell when the operator pulled or pushed the handle. This, in turn, generated a movement output that allowed the handle to follow the surgeon’s hand. The force feedback depended on the speed of the translation stage. The displacement and speed of the translation stage were sent to the catheter manipulator side, which allowed the catheter manipulator to remain synchronized with the controller. When the operator rotated the handle, the rotation angle was measured by an encoder installed on the dc motor. The dc motor functioned in tandem with the control mode to generate a damping force to the surgeon. The damping was calculated from the torque generated by the catheter manipulator side.

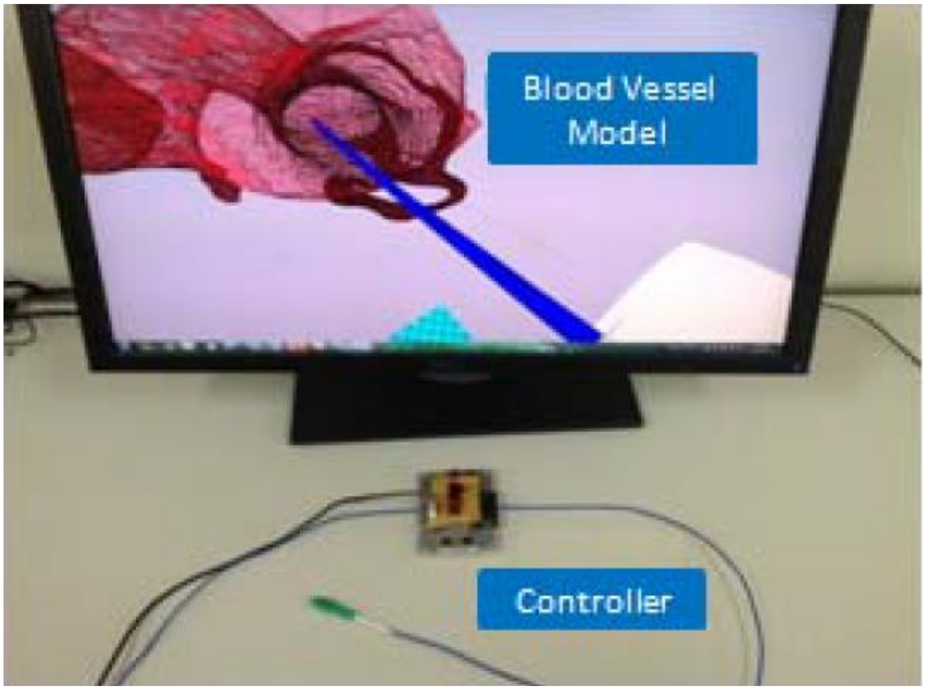


Figure 2- 4 The whole structure of VR-based training system

However, this achievement in this area has not used a real catheter as a controller to operate the virtual vascular interventional surgery. Therefore, it is not convenient when surgeon drive the catheter for inserting and rotating because it does not accord with the custom of surgeons' operations. Therefore, the objective of the optimization effort of the VR-based training system is to present an interventionalist's training system based on virtual reality technology with a new mechanical structure of catheter controller for vascular interventional surgery. The training system can not only generate the realistic virtual reality environment of blood vessels according to patient's special computed tomography (CT) or magnetic resonance imaging (MRI), but also allow unskilled doctors to drive a real catheter for training courses directly and simulate surgeon's operating skills, insertion and rotation in real surgery (shown in Figure 2-4).

The conceptual principle of the controller which was applied to operate virtual catheter inside blood vessel model has been shown in Figure 2-5.

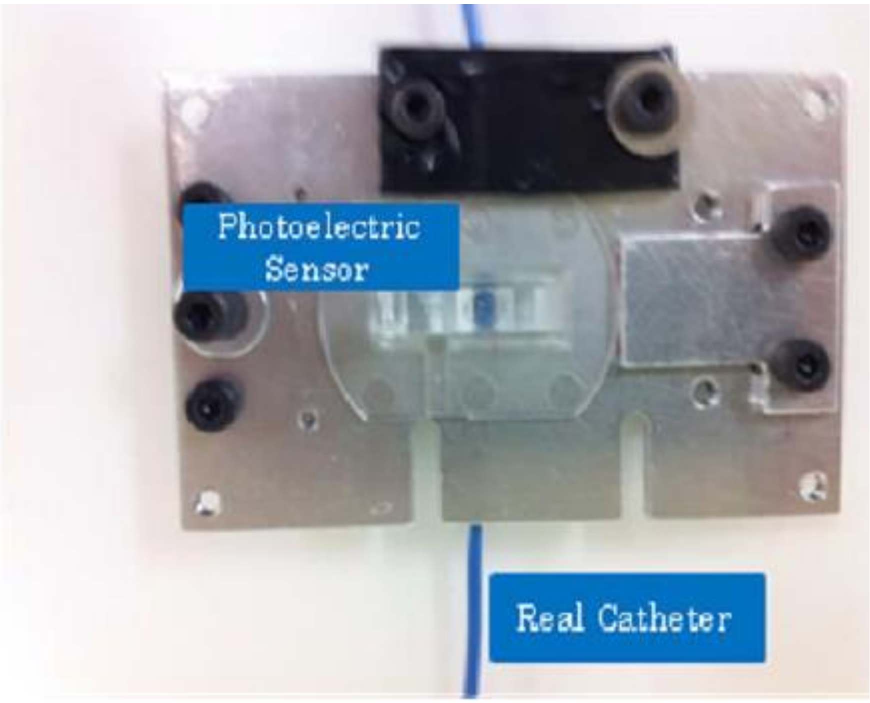


Figure 2- 5 the conceptual principle of the controller

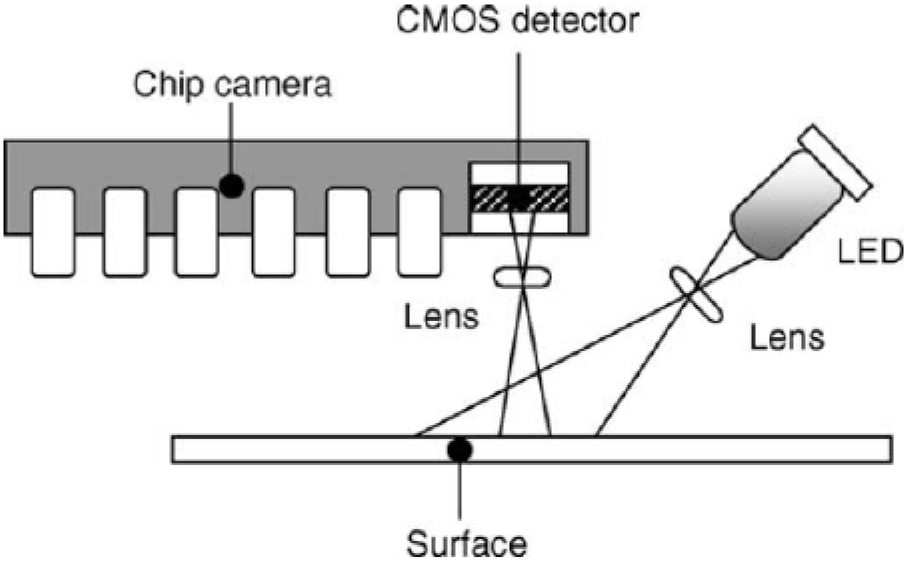


Figure 2- 6 the schematic description of the working components of an optical mouse

The catheter can be subjected to two different sets of movement during manipulation: insertion/retraction and rotation. According to the translation and rotation motion, catheter can be manipulated to reach different parts of the blood vessels. The photoelectric sensor was used to measure the information of displacement and rotation of the catheter. The basic working principle of an optical mouse was described in Figure 2-6. A single light emitting diode (LED) illuminated the surface at an angle. A lens was used to image the surface of the mouse pad onto a CMOS sensor located in the camera chip. The off-axis illumination by the LED helps to put the tiny textures on the surface in sharp contrast. The CMOS sensor typically comprises 18 pixel × 18 pixel (324 pixels in total). The optical sensor works by comparing the images of the surface that are refreshed approximately every 1500th of a second. As it is too computationally taxing to compare the images at all 324 possible overlaps, a 5 pixel × 5 pixel window, taken from the center of the second image, was normally used for the overlap matching process. This window is moved relative to the first image and the chip rates how well each of the 324 pixels matches up. These ratings was added to an overall score for the overlap. Once the chip has found the best overlap, it checks the scores of the eight pixels surrounding the center of the window. Finally, it sended the actual value of the displacement to the computer. Measurement accuracy was typically limited to the pixel spacing of the imaging sensor located in the chip. The unit of data measured by a photoelectric sensor is pixel not millimeter. Therefore, next step was to convert pixel unit into millimeter unit. Some experiments have been carried out to find the relationship between pixel data and real displacement of catheter, as Eq. 2-1 shows:

$$\begin{cases} \boldsymbol{U = a \cdot X} \\ \boldsymbol{\theta = b \cdot Y} \end{cases} \quad \text{(Eq. 2-1)}$$

where X and Y are the sensor outputs in vertical and horizontal direction and U and $\boldsymbol{\theta}$ are real displacements in Cartesian coordinate system. Moreover, a and b are two constants measured by experiments. Then it is much easier to calculate the velocity and acceleration as Eq. 2-2 shows:

$$\boldsymbol{Speed} = \frac{\boldsymbol{U(k) - U(k-1)}}{\boldsymbol{\Delta T}} \quad \text{(Eq. 2-2)}$$

There are two obvious advantages in mechanical design of this controller compared to other training system based on virtual reality technology. The first one is the fact that unskilled doctors can operate the real catheter directly for their training courses and the other one is that the measurement of the translational displacement and rotational angles of the catheter is contactless. Therefore, the whole structure of this controller is simple and has better maneuverability and it is extremely competent to train unskilled interventionalists due to the fact that the operation on this controller is almost the same with the custom of surgeon’s operations in an actual surgery.

2.2 The proposed VR-based training system

The proposed VR-based system for endovascular catheterization training (shown in Figure 2-7 and Figure 2-8) comprises the foundation elements (hardware and software), the core engines (visualization, modeling, and dynamic interaction), and a master controller.

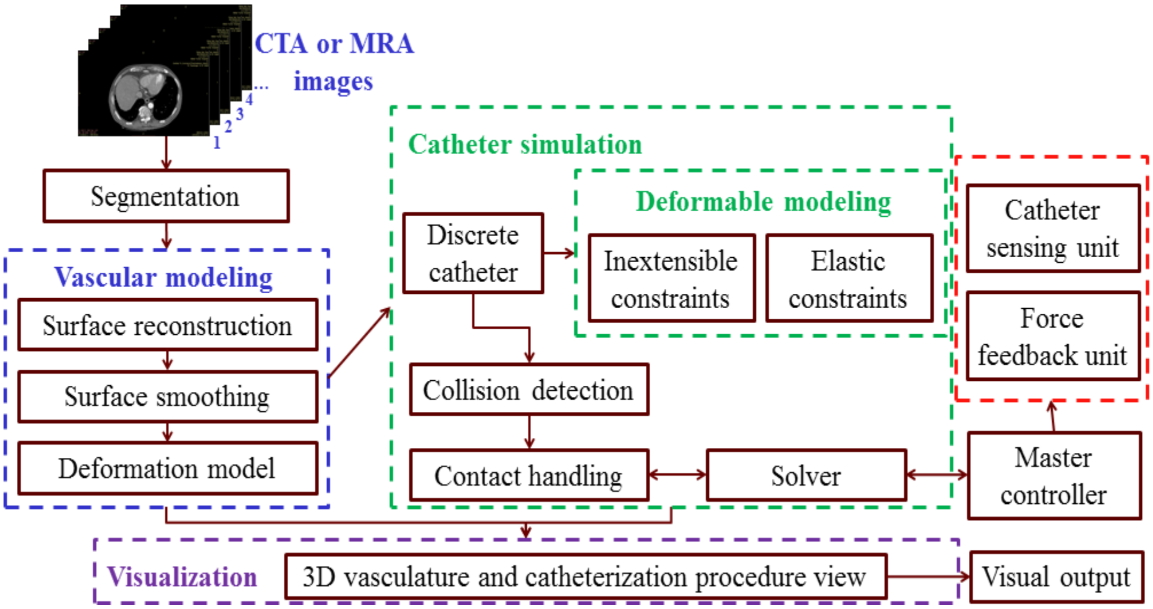


Figure 2- 7 The overview of the proposed VR system for catheterization training

Through the controller, the operator communicates manipulation of the catheter and view to the system, which presents both visual and force feedback. The hardware includes a master controller with force feedback, which will be described in detail later (Chapter 5). Software includes segmentation algorithms for the CTA or MRA images, 3D vascular reconstruction and the physics-based model of the vascular structure and the catheter model.

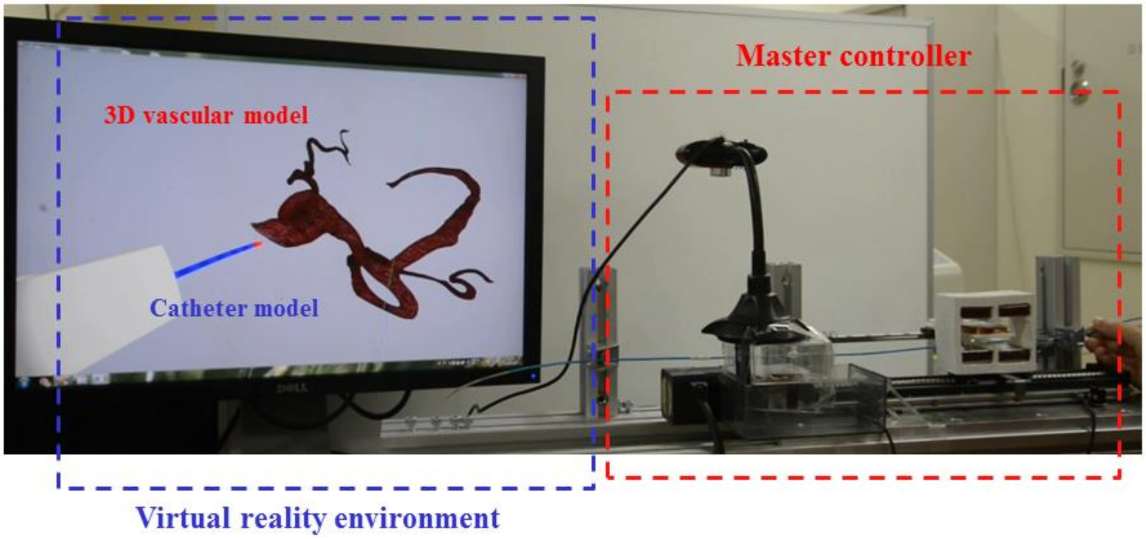


Figure 2- 8 The proposed VR-based system for endovascular catheterization training

Compared to the previous VR-based training simulators, the newly proposed endovascular catheterization training system has many advantages. The deformation models for the catheter and blood vessel walls were added to the virtual reality environment. Additionally, the collision vectors can be calculated and recorded for further analysis. Most importantly, interventionalists were allowed to directly employ an actual catheter with force feedback to carry out the surgical training courses.

2.3 The interaction between the vascular model and the catheter model

During the training procedure, doctors operate the controller to drive catheter to insert or rotate, at the same time, the controlling instructions of the catheter operating system are transmitted to the virtual reality environment. The catheter in virtual reality environment can insert or rotate according to the controlling commands from the controller. Based on the three-dimension vascular information, doctors can decide whether to insert or rotate the catheter.

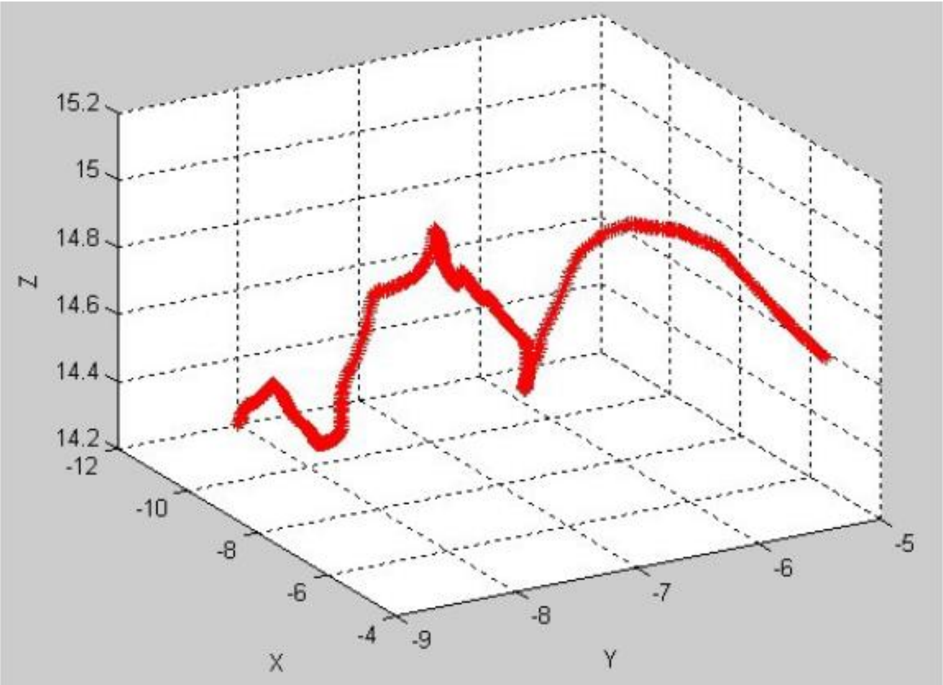


Figure 2- 9 the coordinates of tip of catheter in Figure 2-12

Figure 2-12 hows the status of the tip of catheter which is tracked in a yellow circle inside the blood vessel model when inserting and rotating the controller to control the virtual catheter passing through a curve part and Figure 2-9 shows the three-dimensional coordinates' changes of the tip of the virtual catheter when passing through the curve part.

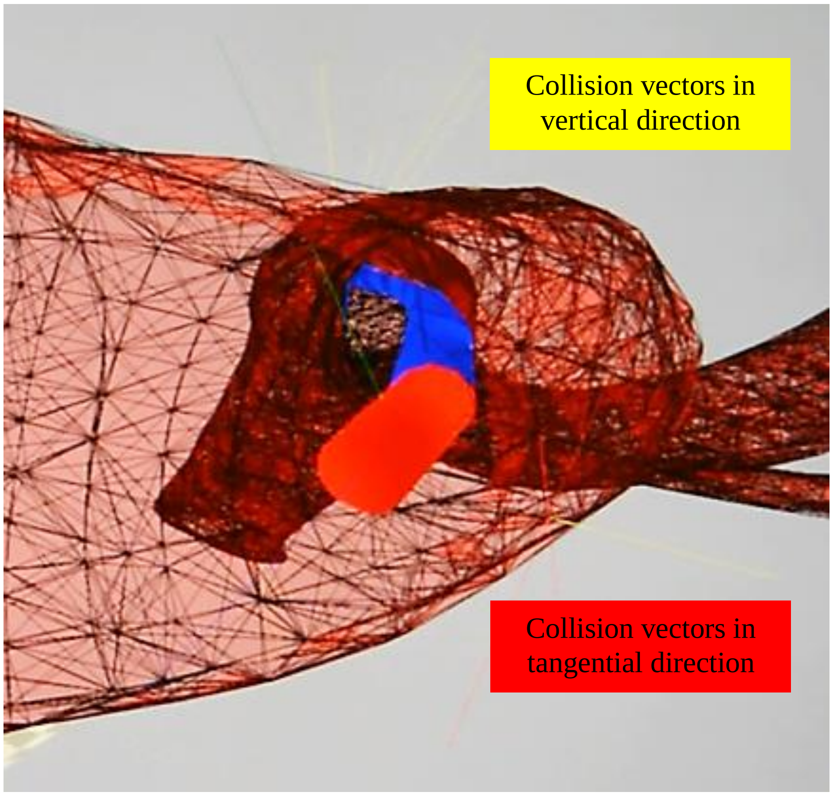

Collision vectors in vertical direction
Collision vectors in tangential direction

Figure 2- 10 The recorded collision vectors

When the collision between the simulated catheter and the blood vessel walls happened, two kinds of vectors, collision vectors in both tangential and vertical directions, were calculated and recorder, as shown in Figure 2-10. The collision vectors of the catheter tip were stored separately with the body part of the catheter, which was used to calculate the tip force feedback (shown in Figure 2-11). In addition, the collision vectors of both the catheter body part and the tip part can be used to calculate the force feedback generated from the virtual reality environment.

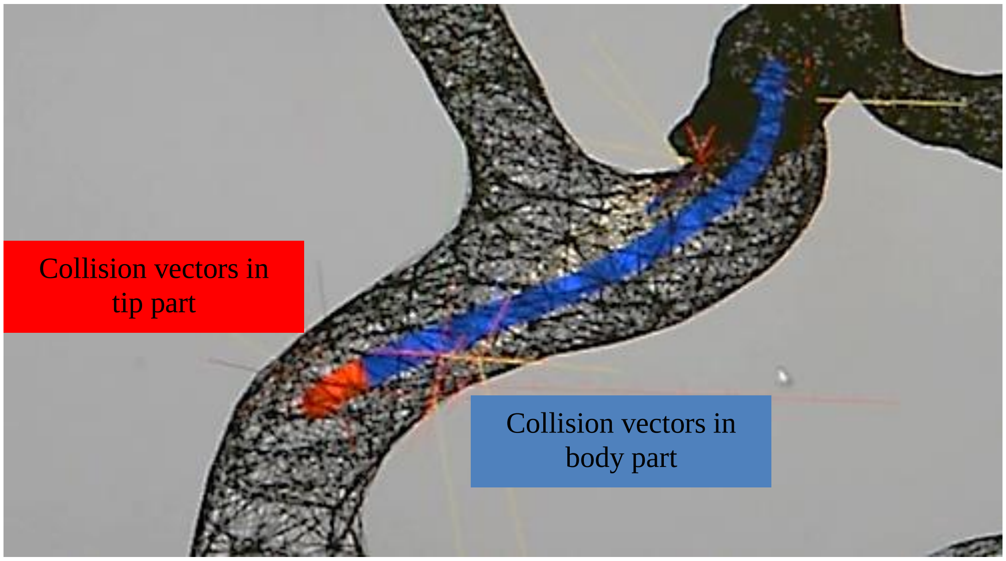
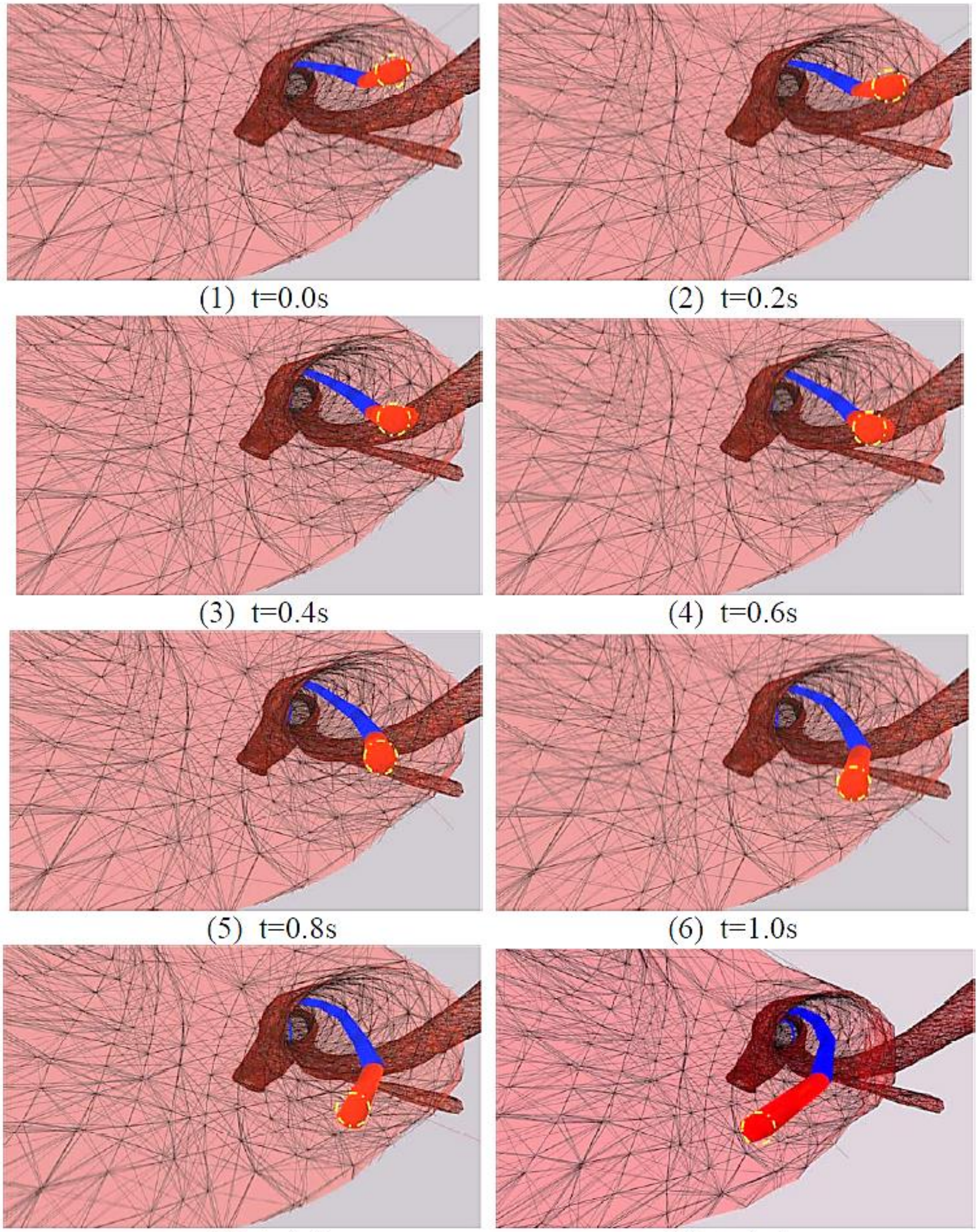


Figure 2- 11 The recorded collision vectors-2

2.4 Summary

In this chapter, the former virtual reality-based training systems have been introduced. Compare to the previous training systems, the overview of the proposed VR-based endovascular catheterization training system was introduced according to the requirements of the software part (virtual reality environment) and the hardware part (the master controller). Finally, the experiments about the interaction between the vascular model and the catheter model were conducted. The collision vectors were calculated and recorded.



(1) t=0.0s

(2) t=0.2s

(3) t=0.4s

(4) t=0.6s

(5) t=0.8s

(6) t=1.0s

(7) t=1.2s

(8) t=1.4s

Figure 2- 12 The Endo Vascular Evaluator model (EVE)

Chapter 3

A hybrid method for 3D vascular reconstruction

3.1 A fast 3D reconstruction method

It is very critical to build the realistic physical-based vascular models that are virtual representations of real blood vessels that display accurate collision detection and deformation. Due to the fact that the reconstructed vascular model was generated based on the specific patients' CT or MRI files, the shape and size were the same with the patients, which results in better training effects to the interventionalists.

In order to reconstruct a realistic three-dimensional vascular model, a median filter algorithm was applied to reduce the noise of CT images, and then use a local thresholding algorithm to realize image segmentation of CT images, finally adopting volume-rendering technology to reconstruct the vascular model.

The median filter is a nonlinear digital filtering technique, often used to remove noise that could be generated in several ways. In this case, noise is generated by the process of image collection. Median filtering is very widely used in digital image processing because, under certain conditions, it preserves edges while removing noise. It is always used in pre-processing steps. The main idea of the median filter is to run through a signal entry by entry, replacing each entry with the median of neighboring entries. This pattern of neighbors is called a "window", which slides entry by entry over the entire signal. Set $\{X_{ij}\ (i, j) \in I^2\}$ determines the grey level of each point of an image, S_{ij} is a filtering window and Z_{ij} means the mid-value of a window in S_{ij}, so

$$Z_{ij}=\mathrm{Med}\{X_{ij}\}=\mathrm{Med}\{X_{((i+r),\ (j+s))},\ (r, s)\in S_{ij},\ i, j\in I^2\} \quad \text{(Eq. 3-1)}$$

The window of the median filter often chooses 3*3, 5*5 or 7*7.

Thresholding algorithms mainly based on regional technology sort image pixels using different characteristic thresholds. The segment algorithm was described as follows: First, set the original image as f (x, y), next find T_1, T_2, T_3……T_N (N >1) according to corresponding algorithms in original images, then sort images into several parts:

$$\text{if } f(x, y)>T_N,\ g(x, y)=L_N;$$

$$\text{if } T_{N-1}<f(x, y)<T_N,\ g(x, y)=L_{N-1};$$

……

$$\text{if } T_1<f(x, y)<T_2,\ g(x, y)=L_1;$$

$$\text{if } f(x, y)<T_1,\ g(x, y)=L_0.$$

Volume rendering methods were employed to generate the images of a 3D volumetric data set without explicitly extracting geometric surfaces from the data. These techniques used an optical model to map data values onto optical properties, such as color and opacity. During rendering, optical properties were accumulated along each viewing ray to form an image of the data. Textures mapping to apply images, or textures, to geometric objects were applied in my case. Volume aligned texturing produces images of reasonable quality, although there is often a noticeable transition when the volume is rotated. The normal and adavanced CT files were shown in Figure 3-3 and Figure 3-4 respectively. Figure 3-5 presents the results of the noise removement.

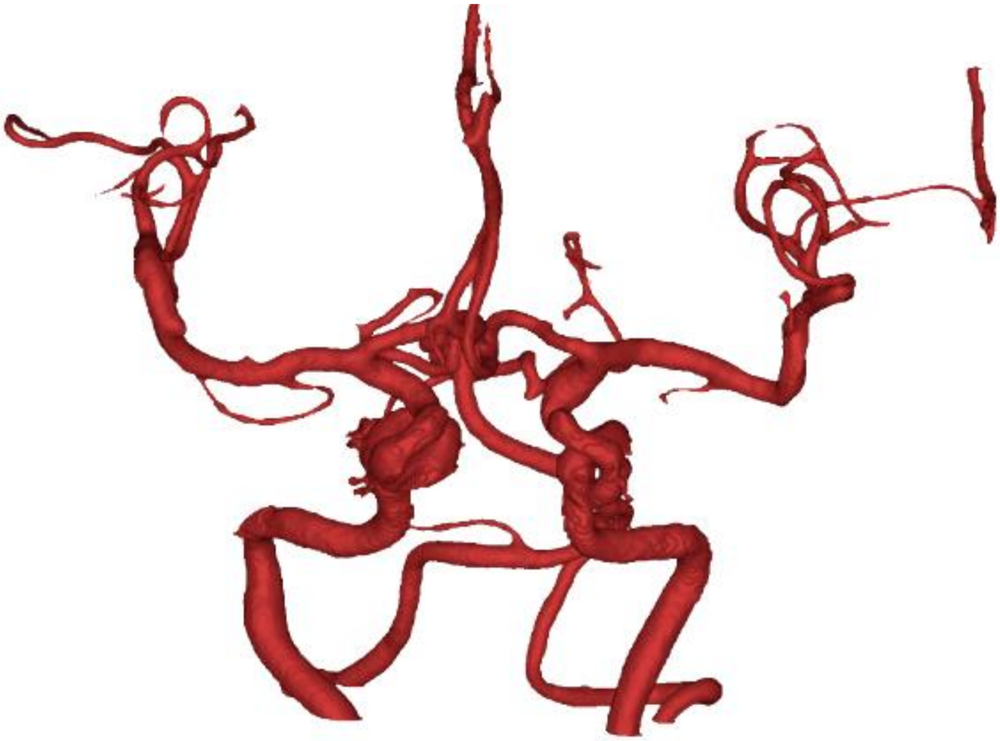


Figure 3- 1 The whole structures of all the blood vessels

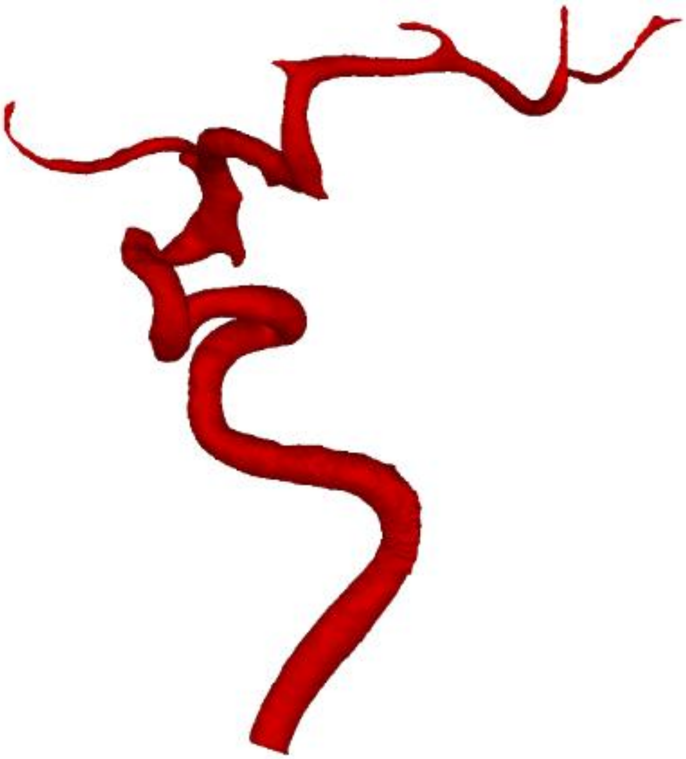


Figure 3- 2 The specific blood vessel

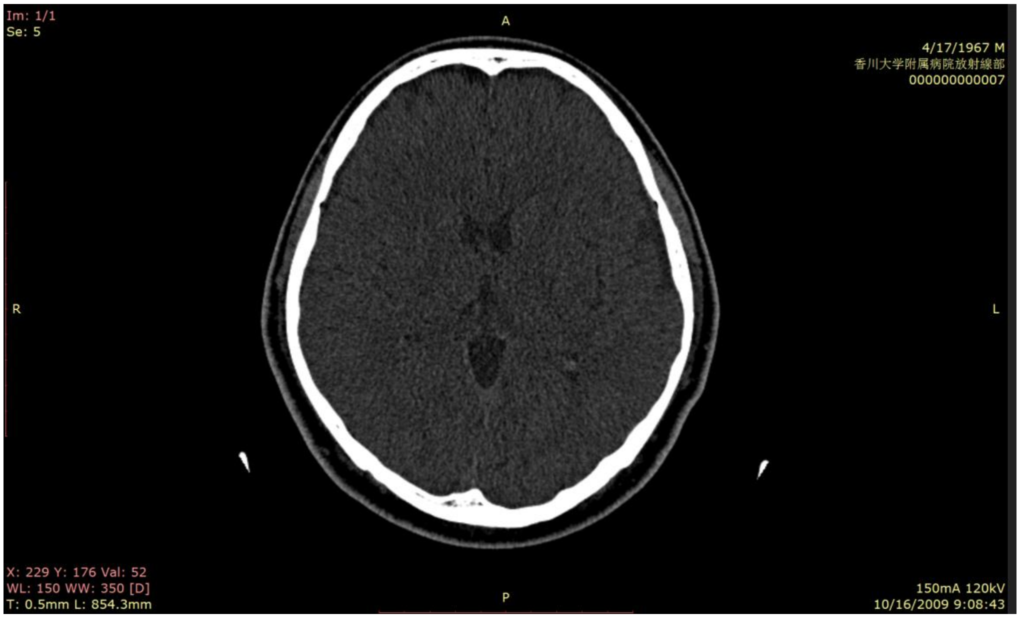


Figure 3- 3 The normal CT file

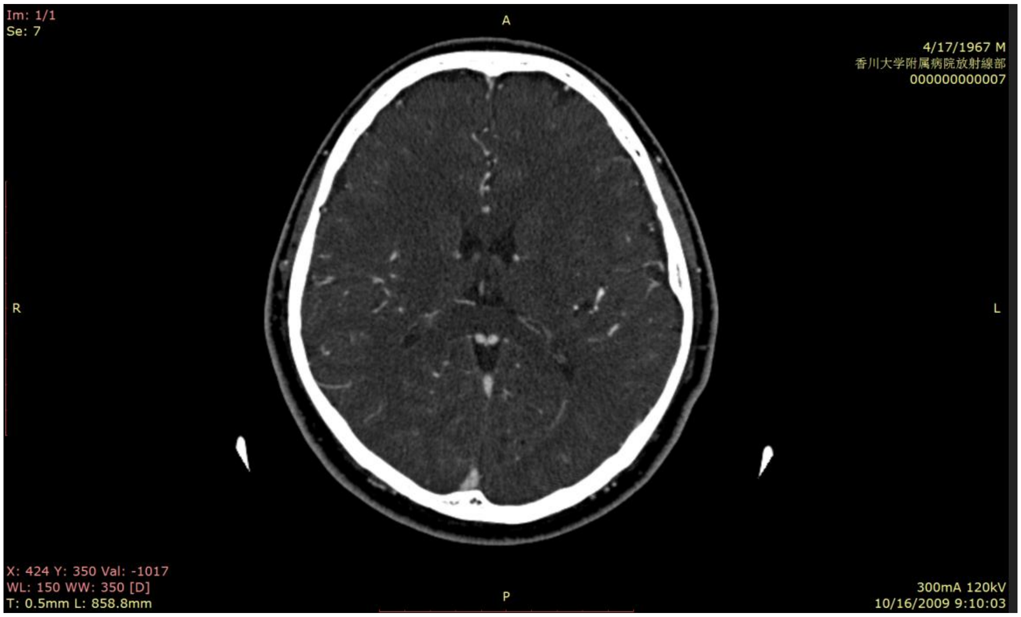


Figure 3- 4 The advanced CT file

Three-dimensional reconstruction images of blood vessels are shown in Figure 3-1 for multi-branched blood vessels and interventionalists can choose part of them as research topic shown in Figure 3-2.

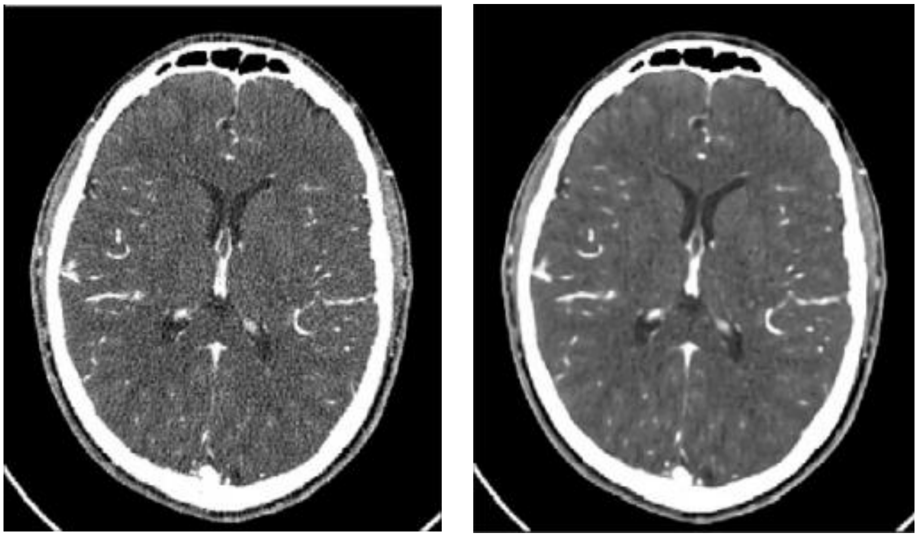


Figure 3- 5 The results of noise removement

If the interventionalists are interested in the blood vessels within the bones or the onely bones, the thresholding values can be adjusted to realize these purposes. The reconstruction results of the blood vessels inside the bones were presented in Figure 3-6, which can better represent the spatial relationship between the blood vessels and the bones. The rebuilt results of the only bones were displayed in Figure 3-7.

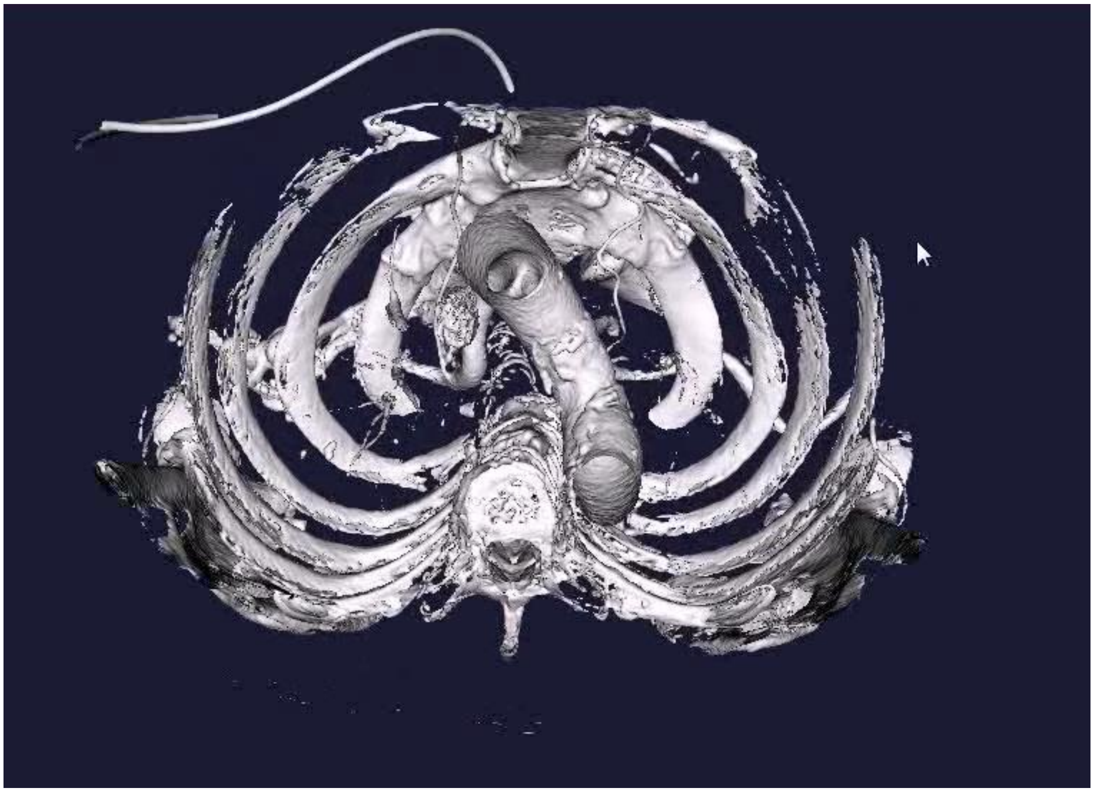


Figure 3- 6 The reconstruction result of the blood vessels inside the bones

3.2 A refined reconstruction method

The reconstructed surface has been integrated into a simulator and fulfills the needs of medical applications: efficient collision detection and collision response between the vascular surface and the simulated surgical tools, physics-based deformation simulation, and anatomical visualization.

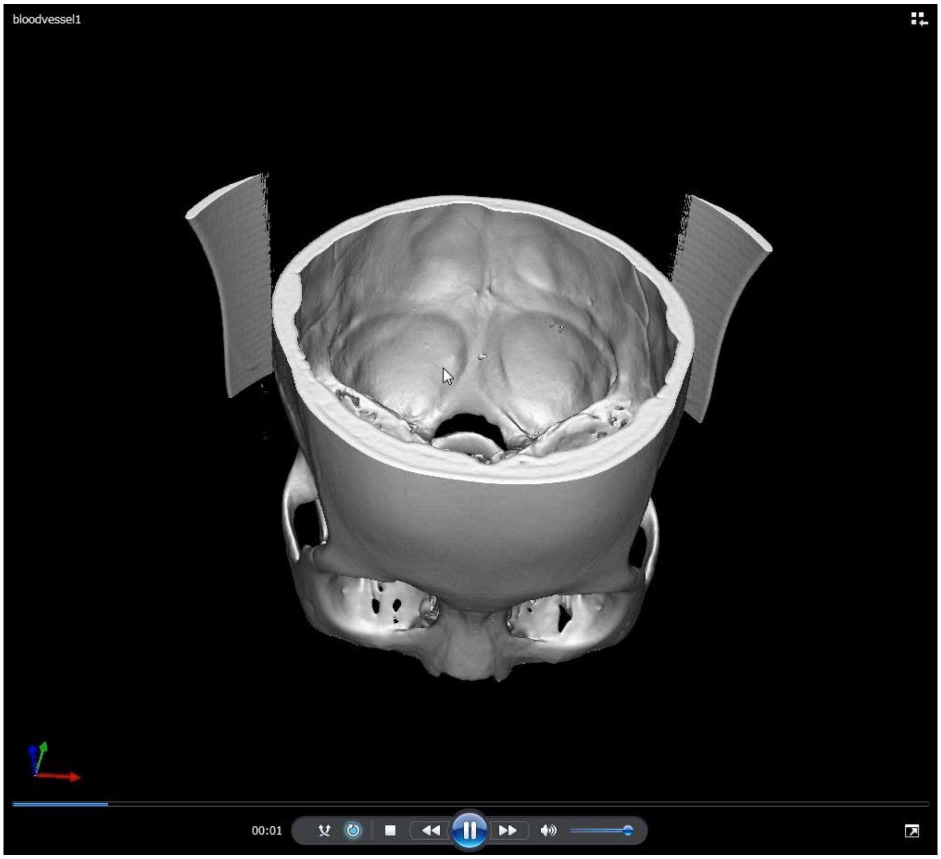


Figure 3- 7 The reconstruction result of the only bones

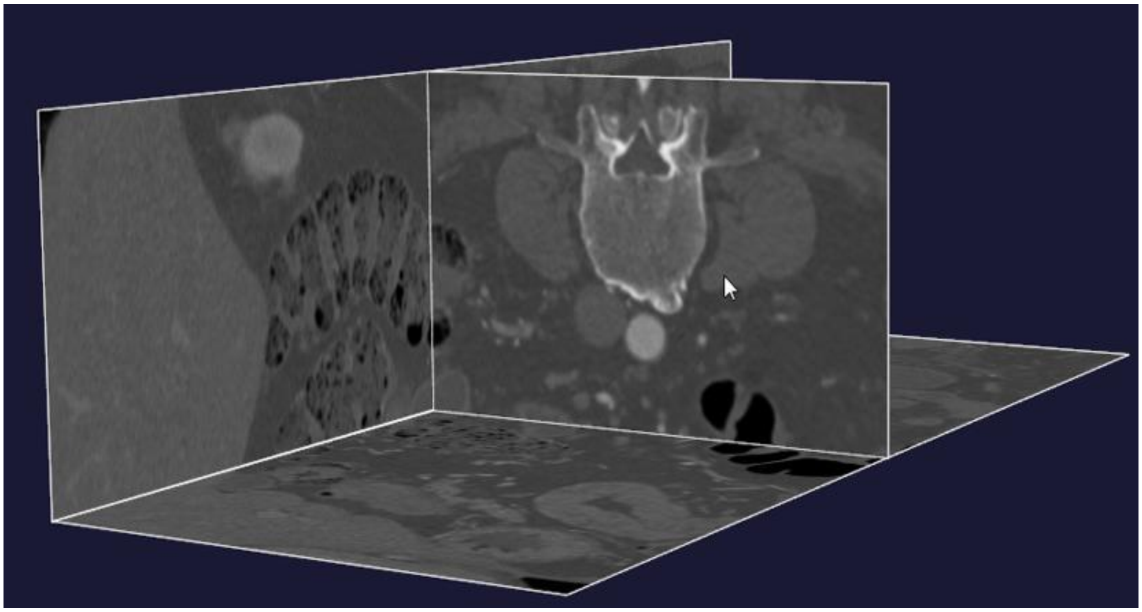


Figure 3- 8 The 3D CT dataset of the specific patient

Different techniques can be used for the segmentation of vascular structures. One of the simplest and sometimes most efficient techniques is thresholding, which usually needs some preprocessing to allow outlining tiny tubular structures. More complex methods, such as centerline enhancement and contour extraction, have more potential for computing the topological representation of a vascular network. Both techniques are very efficient for segmenting the carotids and other relatively large vessels, but have trouble outlining small vessels accurately, especially in the brain when datasets have resolutions coarser than 0.5*0.5*1mm^3. In this study, scans from real patients were used from an electronic database of imaging studies. In this case, the above techniques do not perform well on the whole dataset. Regardless of the technique chosen, it is usually preceded by a noise reduction process, such as anisotropic diffusion filtering. The refined reconstruction technique is composed of the four following steps:

- reduction of the noise of the input MRA or CTA images;
- segmentation of the required blood vessels through a level set evolution, initialized from a manually selected start and end points;
- reconstruction of the blood vessel surface model by using marching cubes algorithm;
- surface refinement.

3.2.1 Enhancement and cleaning

To enhance the medical data set, the first step consists of filtering the images obtained from commercial medical scanners such as Computed Tomography Angiography (CTA) or Magnetic Resonance Angiography (MRA). To reduce the data noise while preserving and enhancing small vascular structures, an anisotropic filter is used.

3.2.2 Level set segmentation

The vessel contours were segmented by means of a level set evolution applied on the enhanced data set. For better efficiency, the active contour was initialized by a threshold on the image intensity. Manual seed selection and region growing methods would be time consuming and less robust since disconnected parts could be missing.

Although the level set technique is computationally more expensive than a straightforward threshold, it allows a better estimation of the contours based on both intensity values and edges. The level set evolves a surface according to three different forces: an advection force that pushes the surface towards the edges of the image; a smoothing term, proportional to the minimal curvature of the surface that keeps the surface smooth; and a balloon force, defined as [Wu11]:

$$F_b(I) = \left\{ \begin{matrix} e^{-\frac{(1-m)^2}{\sigma^2}} - \tau & if\, I < m, \\ 1-\tau \quad if\, I \geq m. & \end{matrix} \right\} \qquad \text{(Eq. 2-2)}$$

The algorithm allows the contours to expand within vascular structures; this force is defined by Eq. 2-2, and depends on the intensity of each voxel. It relies on the intensity statistics to either expand or shrink the evolving contour. I is the intensity image, m stands for the mean intensity of the vessels, r is their standard deviation, and s is a threshold that allows shrinking the contour when the position is unlikely to belong to a vessel. Figure 3-8 shows the 3D CTA dataset. The initial value of Eq. 2-2 with parameters m = 1140, r = 25 and s = 0.2 together with the image intensity histogram after the different pre-processing steps explained in the previous sections.

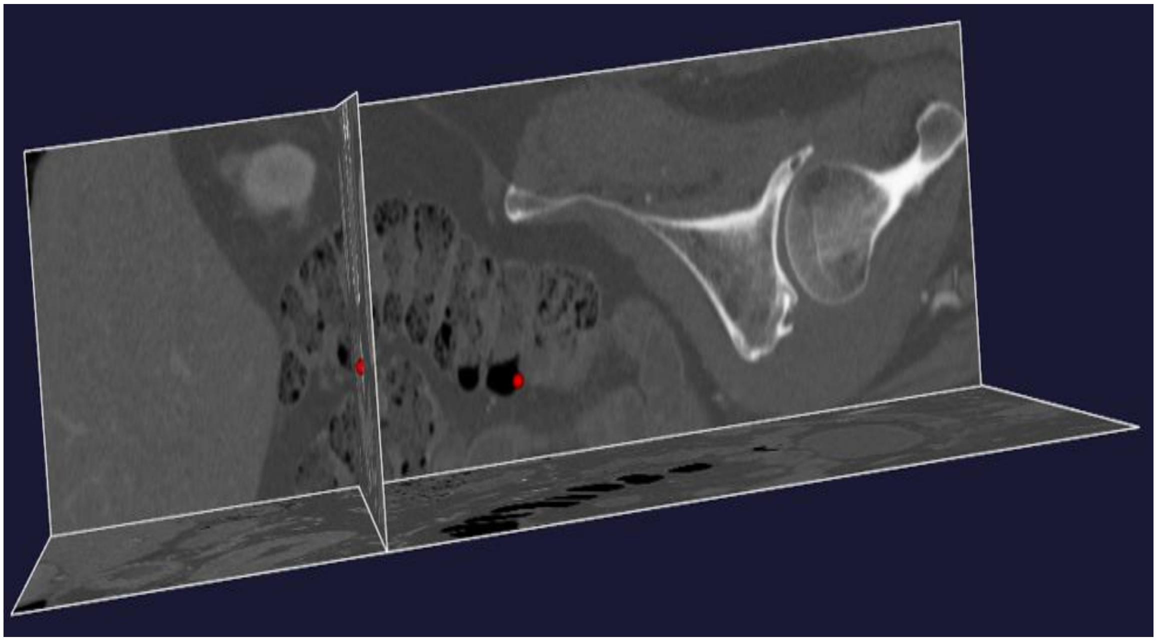


Figure 3- 9 The selection of the start point and the end point of the desired blood vessel branch

The selection of the start point and the end point of the desired blood vessel branches were shown in Figure 3-9. The refined reconstruction algorithm will only deal with the vessel branch from the start point to the end point.

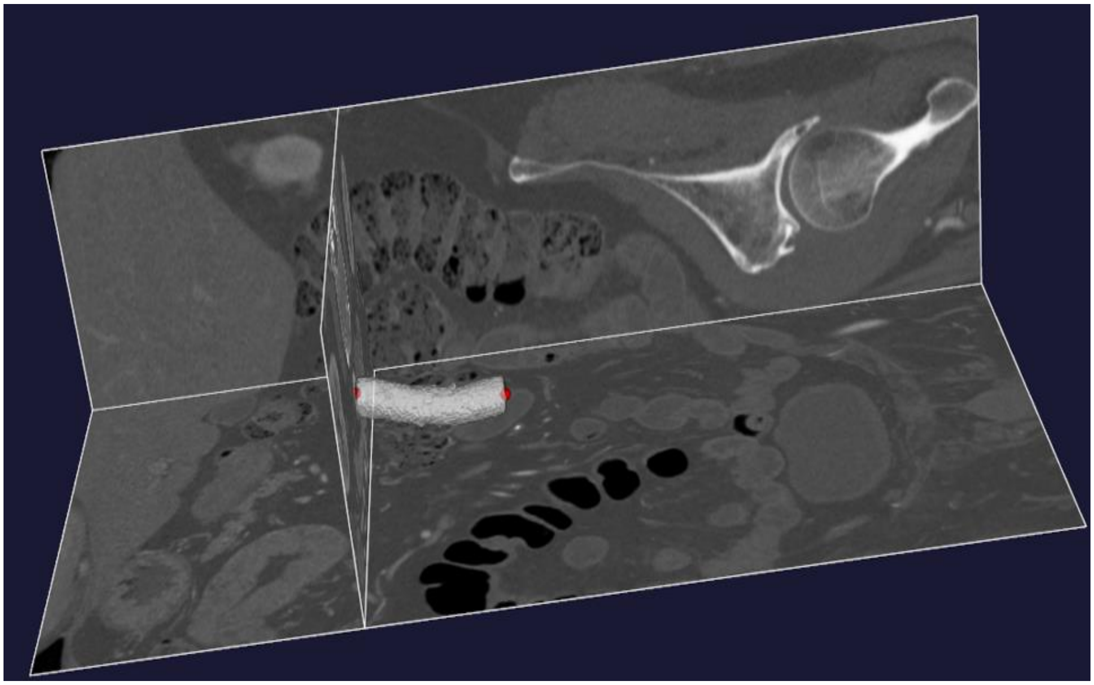


Figure 3- 10 The cloud points' model based on the level sets algorithm

Figure 3-10 presents the cloud points' model after the evolution of the level sets algorithm. Additionally, other desired blood vessel branches can be calculated and integrated to the rebuilt cloud points' model for further analysis. Figure 3-11 shows the selection of the new start point and the new end point. Finally, the newly reconstructed model was integrated with the former one, which was represented in the Figure 3-12. Finally, the whole reconstruction results based on the refined algorithm were shown in Figure 3- 13.

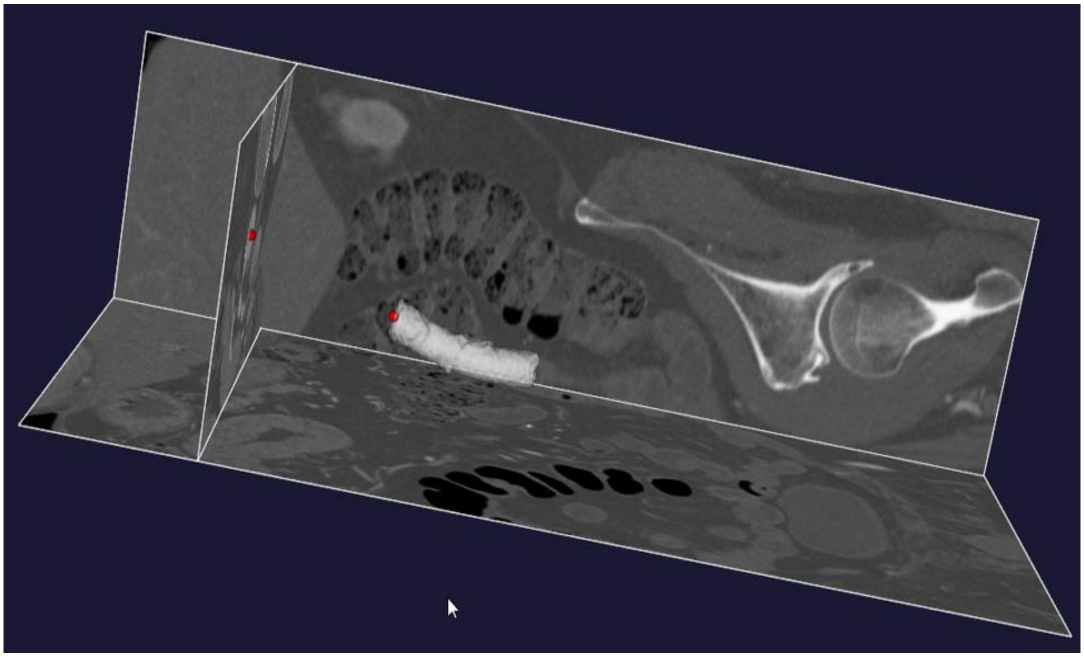


Figure 3- 11 The selction of another segment of the desire branch vessels

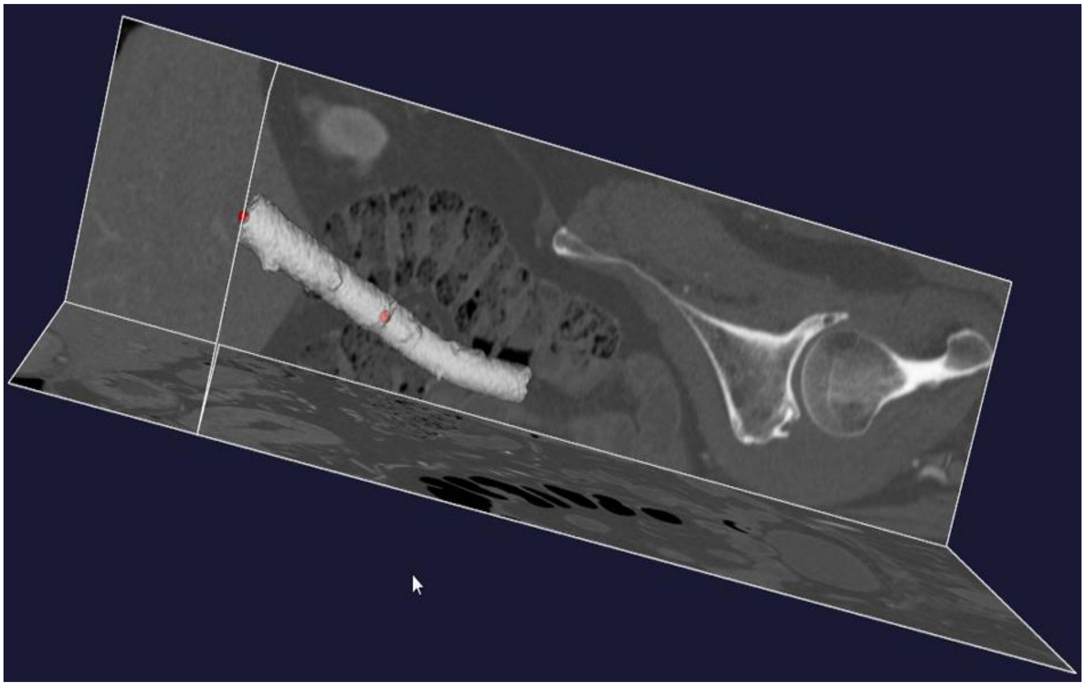


Figure 3- 12 The connection between the two branch blood vessels

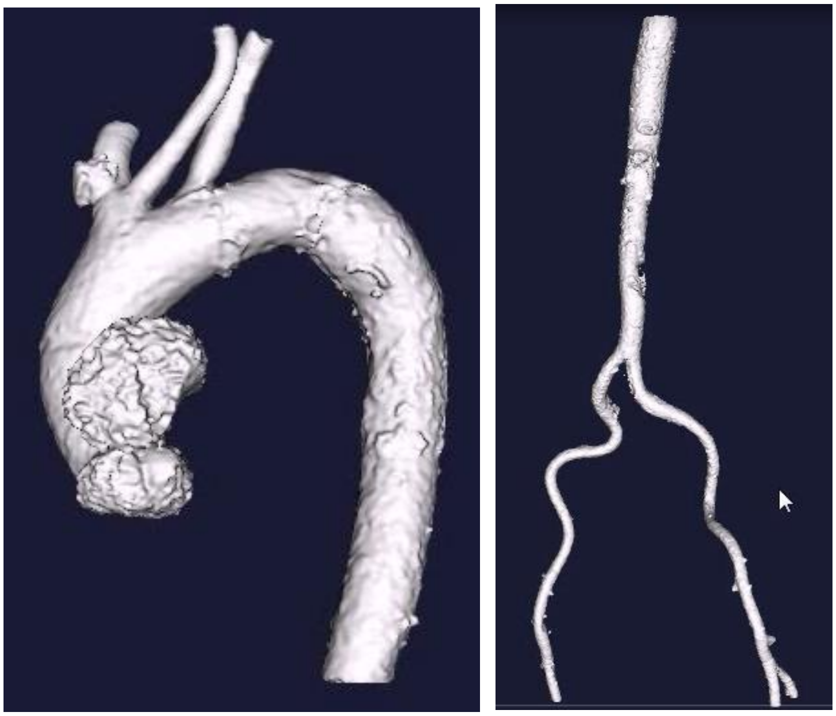


Figure 3- 13 The reconstruction results based on the refined algorithm

3.2.3 Skeletonization and centerline extraction

From the images segmented by the level set equation, a skeletonization technique is applied to obtain the vessels centerlines. This process allows our simulator to efficiently perform collision detection and blood flow computation by supplying an abstract topological representation of the vascular network. The extraction results of the centerlines were shown in Figure 3-14.

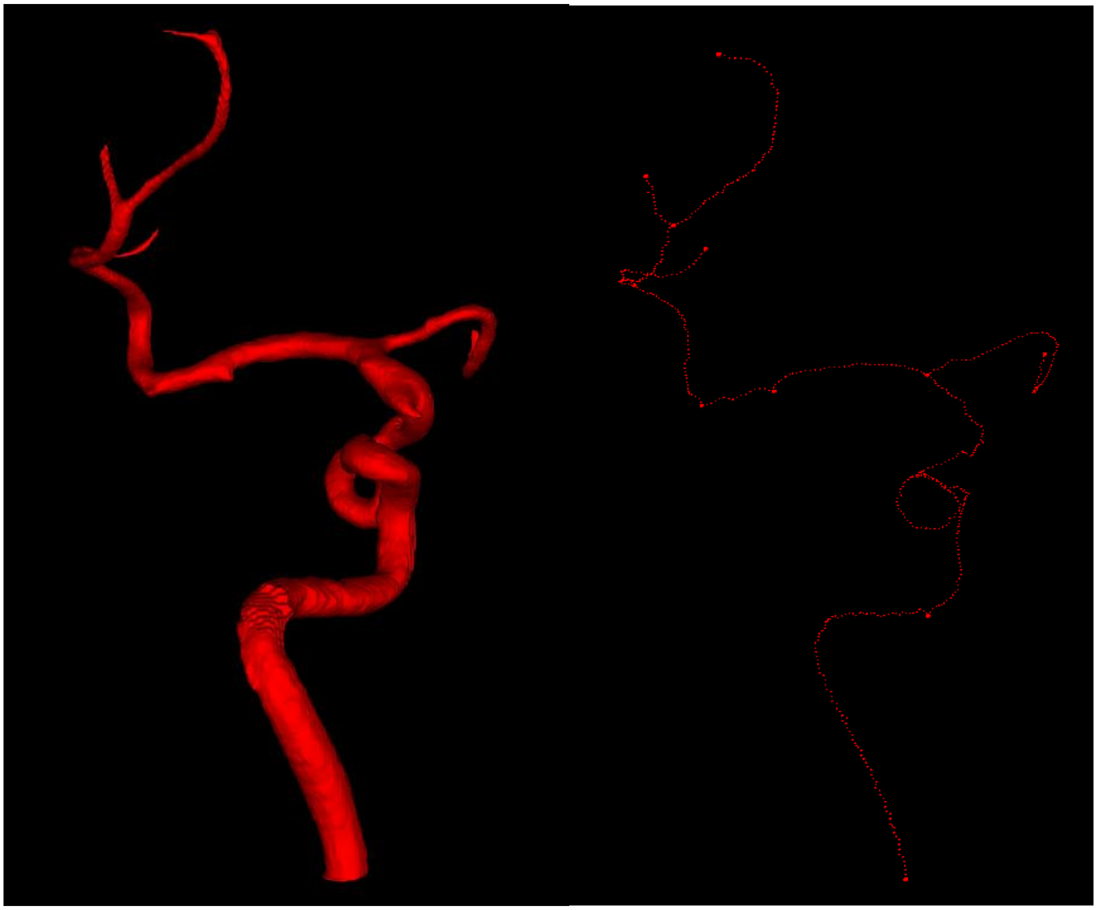


Figure 3- 14 The extraction results of the centerlines

3.2.4 Surface reconstruction

The standard marching cubes (MC) algorithm [Newman06] constructs a facetized isosurface by processing the data set in a sequential, cube-by-cube (scanline) manner. Thus, the approach first processes the m cubes of the first row of the first layer of the data set in sequential order: C_1; C_2; . . . ; C_m. During this processing, each cube vertex V_i that has a value equal to or above the isovalue a is marked; all other vertices are left unmarked. The isosurface intersects each cube edge E_j terminated by one marked vertex V_{js} and one unmarked vertex V_{je}. Any cube that contains an intersected edge is active. The computations that find the active cubes can be viewed as the active cube determination component of MC. This component can be implemented as a stand-alone early processing step or integrated with other processing, but in either case it involves data set traversal in sequential, forward-marching order.

Given an object, a test to determine whether an arbitrary point is within the object, and bounds within which the object exists: Divide the space within the bounds into an arbitrary number of cubes. Test the corners of every cube for whether they are inside the object. For every cube where some corners are inside and some corners are outside the object, the surface must pass through that cube, intersecting the edges of the cube in between corners of opposite classification. Draw a surface within each cube connecting these intersections.

Since each of the eight vertices of a cube can be either marked or unmarked, there are 256 possible marking scenarios for a cube. Each cube marking scenario encodes a cube-isosurface intersection pattern (i.e., configuration). However, the standard MC considers reflective and rotational symmetry, which results in just 15 marking scenarios. Cubes C and $\bar{C}$ are reflectively symmetric if each vertex at position V_i in C has the opposite marking as the vertex at the same position V_i in $\bar{C}$. Two reflectively symmetric cubes (and their cube－isosurface intersections) are shown at the center and right (i.e., cubes A and AF) in Figure 3-15. In the figure, circle symbols denote marked vertices. Cubes that are reflectively symmetric have the same cube－isosurface intersection pattern. Two cubes C and $\bar{C}$ are rotationally symmetric if there is some series of rotations R which, when applied to C, transforms C to a new orientation in which the marking at each transformed vertex position Vi is identical to the marking at the same position Vi in $\bar{C}$. The cubes A and AR in Figure 3-15 are rotationally symmetric. Rotationally symmetric cubes have equivalent cube－isosurface intersection patterns.

The 15 unique cube－isosurface intersection scenarios that result when considering both of these symmetries are shown in Figure 3-16. I will use the topological case numbering of Figure 3-16, which is the same as the standard MC, throughout this survey. For each scenario, the standard MC facetization of the intersecting isosurface is shown. Lorensen and Cline stored facetization information (specifically, the vertices of the triangle(s) to be generated, each identified by the edge on which it lies) about the 15 intersection topologies in a look-up table built offline prior to application of MC. The isosurface–edge intersection locations can be estimated with subvertex accuracy using an interpolation technique. Standard MC employs linear interpolation to estimate the intersection point for each intersected edge. If a unit-length edge E has end points V_s and V_e whose scalar values are L_s and L_e, respectively, then given an isovalue a, the location of intersection I= (I_x; I_y; I_z) has components of the form:

$$I_{\{x,y,z\}} = V_{S\{x,y,z\}} + \rho(V_{e\{x,y,z\}} - V_{S\{x,y,z\}}) \qquad \text{(Eq. 2-3)}$$

The last step in MC is to generate triangular facets that represent the portion of the isosurface that intersects each cube. The intersection points define the vertices of the triangles, and the collection of the triangular facets across all the cubes forms the triangular mesh (or meshes) that defines the isosurface. The facetization pattern in each cube can be determined from the intersection topology look-up table. The processing steps that build the facetization can be viewed as the isosurface assembly component of the MC.

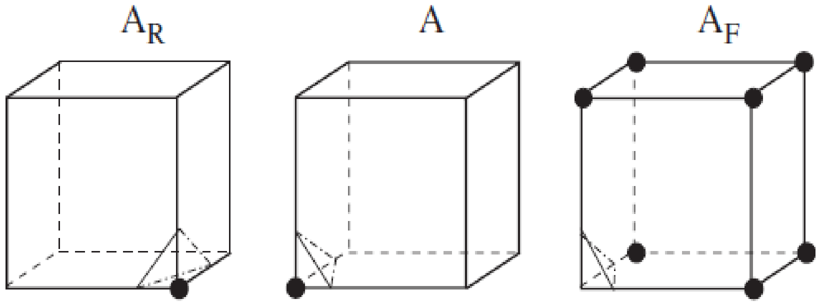


Figure 3- 15 The illustration of reflective (A with AF) and rotational (A with AR) symmetries.

The surface reconstruction based on the marching cube algorithm was shown in Figure 3-17 and the rendering results of the surface model after the surface refinement was presented in the Figure 3-18. Additionally, the surface refine work was conducted and the results were shown in Figure 3-19.

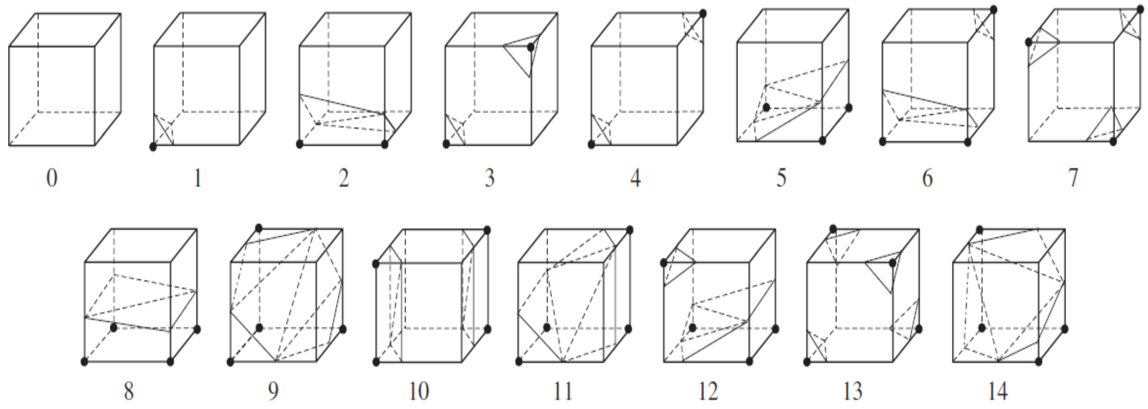


Figure 3- 16 The 15 basic intersection topologies

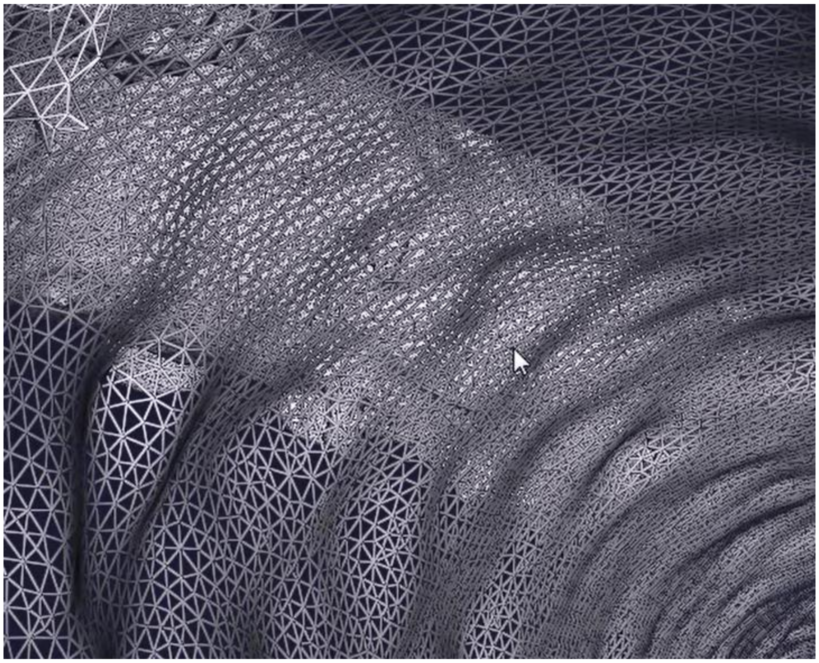


Figure 3- 17 The surface reconstruction based on the marching cube algorithm

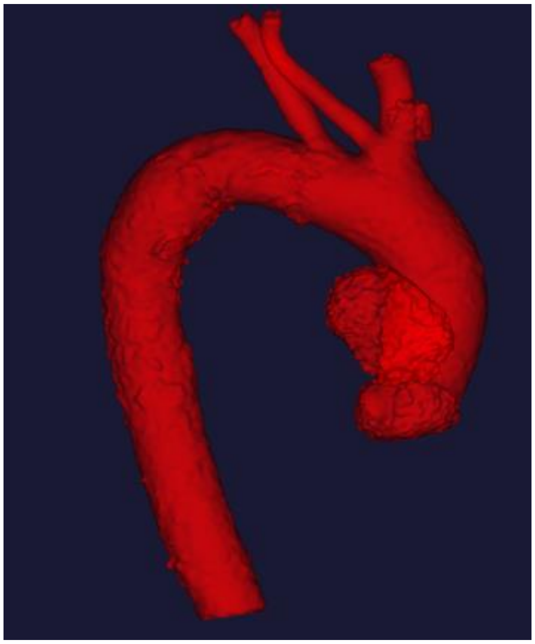


Figure 3- 18 The rendering results of the surface model after the surface refinement

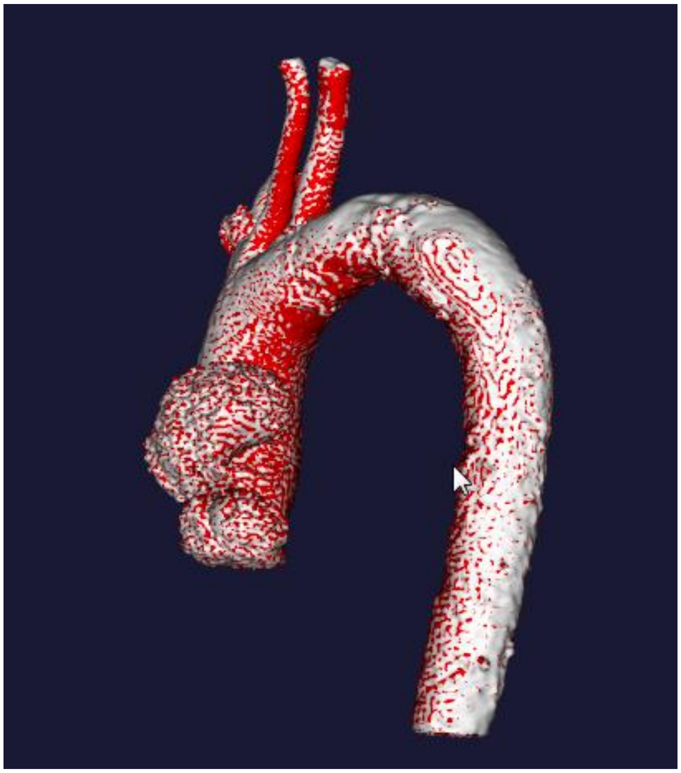


Figure 3- 19 The comparison results before and after surface refinement

3.3 Summary

In this chapter, a hybrid method for 3D vascular reconstruction was introduced. The fast 3D reconstruction method was used based on the normal and advanced CT or MRI files. The main advantage of this method is the fact that it can reach a high speed to rebuild the 3D vascular model. The other reconstruction method can be employed only based on the CTA or MRA files. It includes six steps to reconstruction the vascular model. The main advantage of the refined method is its flexibility. It can generate the blood vessel model which is desired by the interventionalists. Obviously, it is more time-consuming than the fast reconstruction method.

Chapter 4

Physics-based models for the virtual reality-based training system

4.1 Physics-based modeling of blood vessels

There are two main effective approaches for simulating blood vessels precisely-- finite element method (FEM) and the mass-spring method (MSM). Compared to the FEM, the MSM presents better performance in calculation speed and adaptation to the topological structure of blood vessels. The MSM is also precise enough in the simulation of blood vessels, which is demonstrated in reference [Zhou10]. I therefore used the MSM to construct the physic model of blood vessels.

The mass-spring model is a mesh-free method widely used in surgical simulation [Ye11], which models an object as masses mutually connected with springs and dampers. Each mass is represented by its own coordinates, acceleration and velocity and is deformed under the influence of inertial, spring and damping force and force applied by the surgical catheter. This approach is relatively easy to implement because the motion equations used need not be constructed explicitly, hence, real-time simulation is achieved due to the technique’s computational complexity. More importantly, the density of the mesh according to precision and real-time requirements can be adjusted.

In the simulators, the spring-damper model was applied to calculate force feedback [Zhou10]. First, the nearest mass was obtained as the base point of the deformation and force calculation. The method we chose was to count the distance between the contact point and near discrete masses, then select the mass with the shortest distance to a contact point.

Hooke’s law describes this force by a "spring equation" (1). F is the resulting force, k is the stiffness of the spring, lij is the length of the spring connecting i-th and j-th particle while the zero superscript again denotes the rest pose.

$$\boldsymbol{F}_{\boldsymbol{i}}^{\boldsymbol{int}} = \sum_{\forall \boldsymbol{p}_{\boldsymbol{j}} \in \boldsymbol{S}_{\boldsymbol{i}}} \boldsymbol{k} \cdot \left(\left| \boldsymbol{P}_{\boldsymbol{j}} - \boldsymbol{P}_{\boldsymbol{i}} \right| - \boldsymbol{a} \right) \cdot \frac{\boldsymbol{P}_{\boldsymbol{j}} - \boldsymbol{P}_{\boldsymbol{i}}}{\left| \boldsymbol{P}_{\boldsymbol{j}} - \boldsymbol{P}_{\boldsymbol{i}} \right|} \qquad \text{(Eq. 4-1)}$$

The movement of the particles can be described by Newtonian mechanics. When only one spring and one particle is accounted for, it takes the form of equation (2), where m is the mass of the observed particle, c is the damping coefficient of the spring, k is again the stiffness coefficient and x is the position of the particle, with appropriate time derivatives.

$$m\ddot{x} + c\dot{x} + kx = 0 \qquad \text{(Eq. 4-2)}$$

Eq. 4-3 shows the actual form that needs to be solved for every particle i in a general MSS, with F_e representing external forces acting on the particle, F_{ij} the force computed using equation (1) and N_i the set of particles, to which particle i is connected by a spring.

$$m_i a_i + c_i v_i + F_i^{int} = F_i^{ext} \qquad \text{(Eq. 4-3)}$$

To obtain an exact solution of the differential equation (Eq. 4-3), it has to be integrated in time. Various integration schemes have been tested and Verlet integration emerged as the most suitable for application in MSS. Moreover, it is quite simple to implement. It discretizes time by replacing the derivatives by differences between sufficiently small steps dt. The step dt is an additional parameter of the system which contributes heavily to its behaviour – a too small step will result in lengthier computations while too big steps will result in divergence of the integration scheme and therefore the system itself (i.e. it will not be able to achieve a stable position).

4.2 Physics-based modeling of the catheter

The approaches of the catheter simulation have been presented by several research groups. The algorithms can be classified as physical or geometrical methods. Geometrical methods, such as splines and snakes, are based on a simplified physical principle to achieve the simulation results. Thus, calculation rate of the virtual model using this algorithm is fast but without physical properties. The main physical approaches to catheter modeling are the mass spring modeling (MSM) methods. It describes a shape as a set of basic geometrical elements and the model is defined by the choice of its elements, its shape function, and other global parameters. So the MSM is a suitable technique for solving the simulation problem. Based on the catheter structure, the catheter was discretized as a chain of small and elastic cylindrical segments, as shown in Figure 4-1 (a). Each one was connected to its neighbors at joints known as nodes. The small cylindrical segment was also called the beam element. Two successive beam elements form one bend element. With these elements we can evaluate the deformation energy and the elastic force of the structure. The virtual catheter was shown in Figure 4-1 (b).

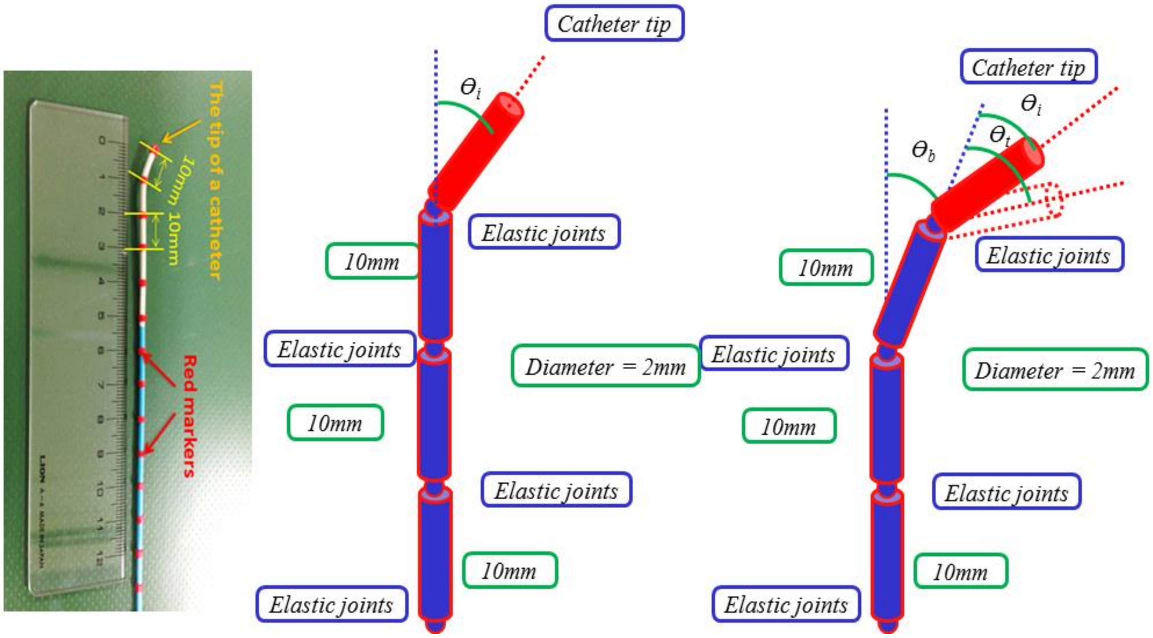


Figure 4- 1 The simulation images of catheter: (a) the real catheter. (b) the simulated catheter in VR system

The elastic joint based method was used to realize the physical model of the catheter because this virtual model runs more accurate and stable in simulation of catheter. In order to obtain the physical characters of the catheter, the operation of insertion and rotation should be considered, when the catheter is driven into the blood vessel.

When the segment element is stretched or compressed, it generates elastic forces on its nodes. The direction of the force is in the direction of segment element and its magnitude follow Hooke's law. The spring force constrains the stretching of the catheter

$$F_{spring} = \sum -k_s \left(l_{\{i-1,\, i+1\}} - l\right) \cdot v_{\{i-1, i+1\}} \qquad \text{(Eq. 4-4)}$$

where *ks* is the spring force coefficient and $l_{\{i-1,\, i+1\}}$ is the distance between the particle *i* and its neighbors, and $v_{\{i-1, i+1\}}$ is the vector between the particle *i* and its neighbors [Luboz09].

Additionally, the bending force keeps the original shape of the catheter:

$$F_{tip-bending} = \; -k_b(\theta_t - \theta_i) \cdot \boldsymbol{u_t} \qquad \text{(Eq. 4-5)}$$

$$F_{body-bending} = \; -k_b\theta_b \cdot \boldsymbol{u_b} \qquad \text{(Eq. 4-6)}$$

4.3 Collision detection

Apart from the models themselves, there is another important topic when considering training based on virtual reality technology and that is collision detection and response. The difficulties bounded with soft bodies such as blood vessel walls stem from their complicated reactions to external influences. In this case, two kinds of collision detection methods are applied, broad phase and narrow phase. In terms of broad phase, bounding volume hierarchies (BVHs) are probably the most popular mechanisms. The idea is to recursively subdivide the object of interest and compute a bounding volume for each of the resulting subset of primitives. Then, when checking for collisions, the hierarchy of the potentially colliding pair of objects is traversed from top to bottom. During the traversal, the bounding volumes are tested for overlap on every subdivision level. If no overlap is found, the objects surely cannot collide. If it is, the algorithms traverse the hierarchy further, but only through the children nodes where an overlap was detected. Finally, when the traversal gets to the bottom level of the hierarchy and still detects overlaps, the primitives stored in these nodes are finally tested for mutual intersection. Actually, any BVHs are generally a good solution for complex scenes because they are easily used for self-intersection tests and quite convenient. As for the narrow phase, two stages are performed for collision detection, first checking to see if any vertex in the blood vessel model lies within the catheter model then again checking if a vertex of the catheter model lies within the virtual vascular model. Since every triangle stores an outwardfacing normal ni, a point on the triangle ti(this can be any one of the vertices), and given a colliding vertex v, a vertex can be said to be inside an object if for every triangle in the rigid object (***p-ti***)·***ni*** <0. The minimum collision depth is just the largest scalar value from (***p-ti***)·***ni*** if the intersecting test passes for all triangles.

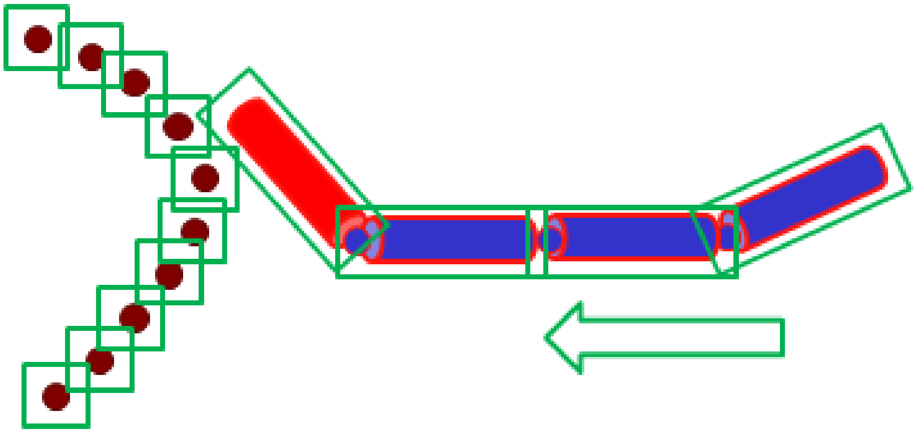


Figure 4- 2 The collision detection between the simulated catheter and the blood vessel walls

4.4 The performance evaluation of the deformation model of the blood vessel walls and the simulated catheter

During the training procedure, interventionalists manipulate the real catheter to drive the simulated catheter inside the virtual reality environment to insert or rotate, at the same time; the controlling instructions of the catheter operating system were transmitted to the virtual reality environment. When there is a collsion between the blood vessel walls and the simulated catheter, the collision detection algorithms were triggered to record the collision vectors and then the vascular structure of the virtual catheter were deformed according to their physics-based models respectively. Based on the three-dimension vascular visual feedback, interventionalists can decide whether to insert or rotate the catheter. Figure 4-3 presents the vectors attached to the vertices of the blood vessel walls. The deformation of the blood vessel walls was displayed in Figure 4-4. Additionally, Figure 4- 3 shows the deformation of the simulated catheter.

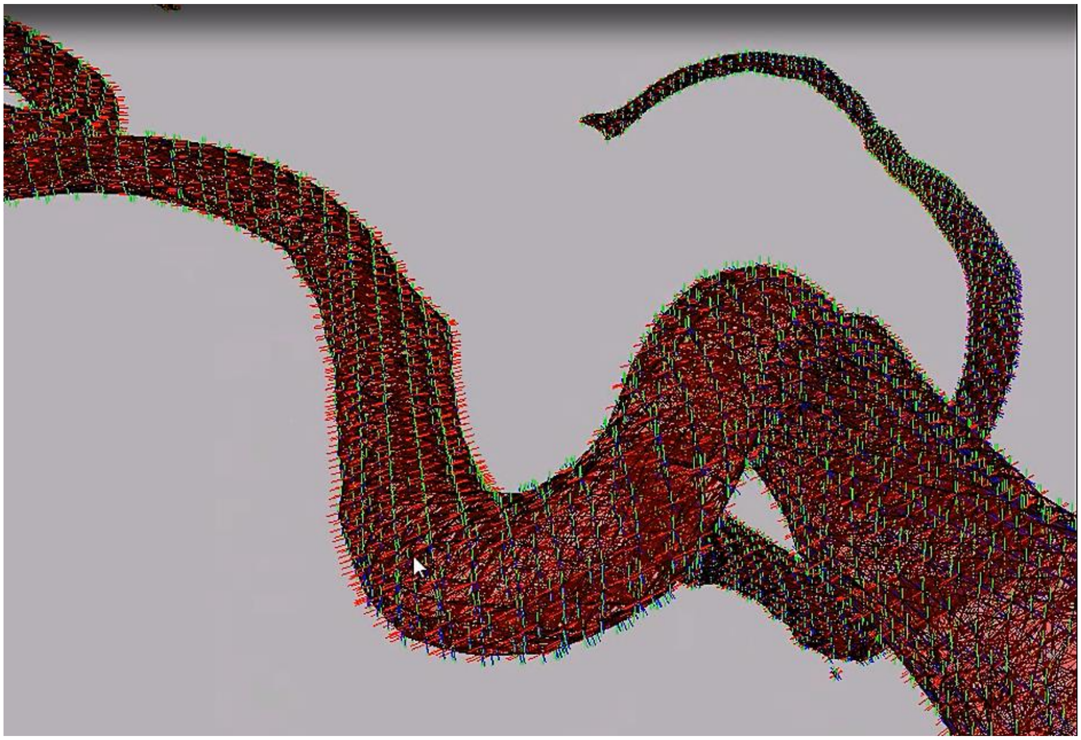


Figure 4- 4 The vectors attached to the vertices of the blood vessel walls

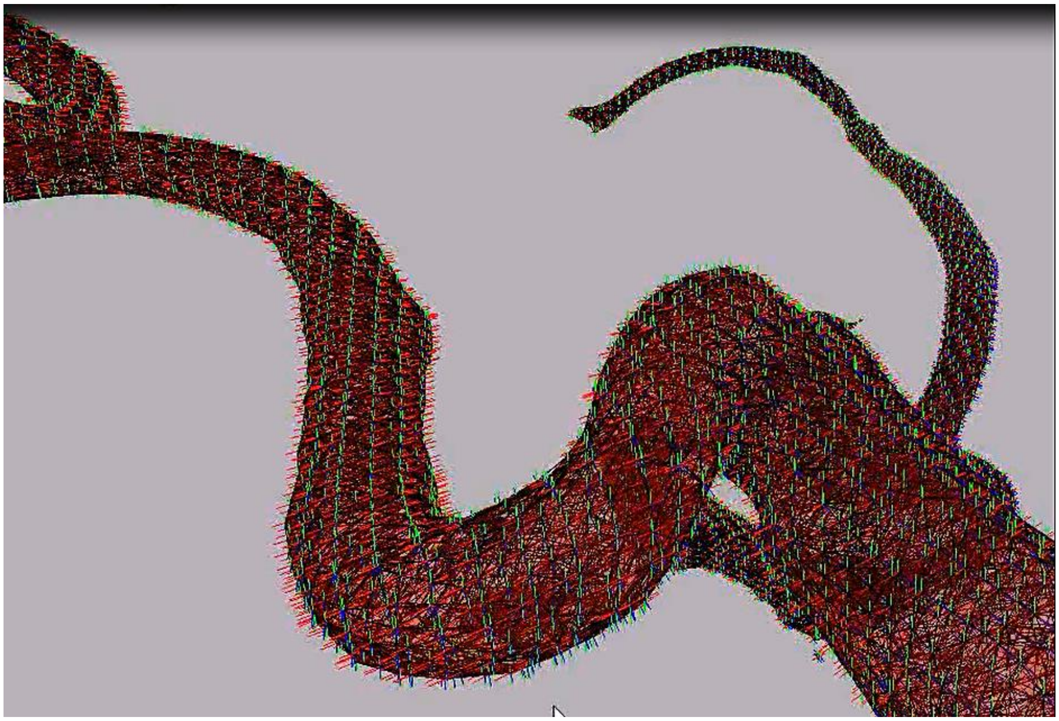


Figure 4- 5 The deformation of the blood vessel walls

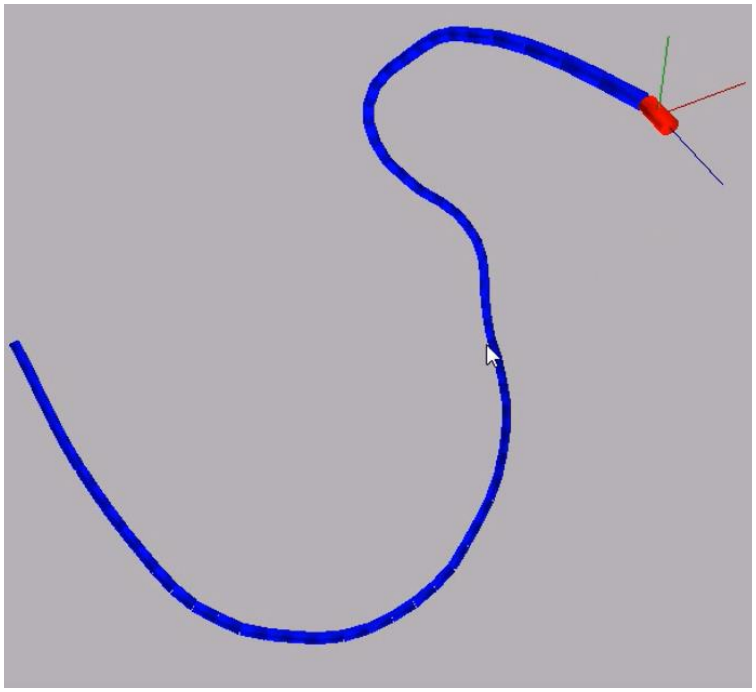


Figure 4- 6 The deformation of the simulated catheter

4.5 Summary

The result of the work presented here were virtual-reality simulators allowing the simulation of a surgeon's operations for training in minimally invasive surgery. Based on the virtual model of the catheter and blood vessels, we have analyzed and applied physical-based theory and implementation for these models. Experimental results have shown that by defining the material properties of the catheter and blood vessels, the behavior of catheter motion is realistically simulated in a specific patient's artery network, thereby allowing surgeons to train and rehearse new operative skills repeatedly.

A simple mass spring model was employed to simulate the deformation behavior of the blood vessel wall in order to obtain the real-time computation. A elastic joints based model was introduced to simulate the deformation behavior of the catheter. The collision detection method was realized and the collision information was recorded. The interaction experiments were conducted.

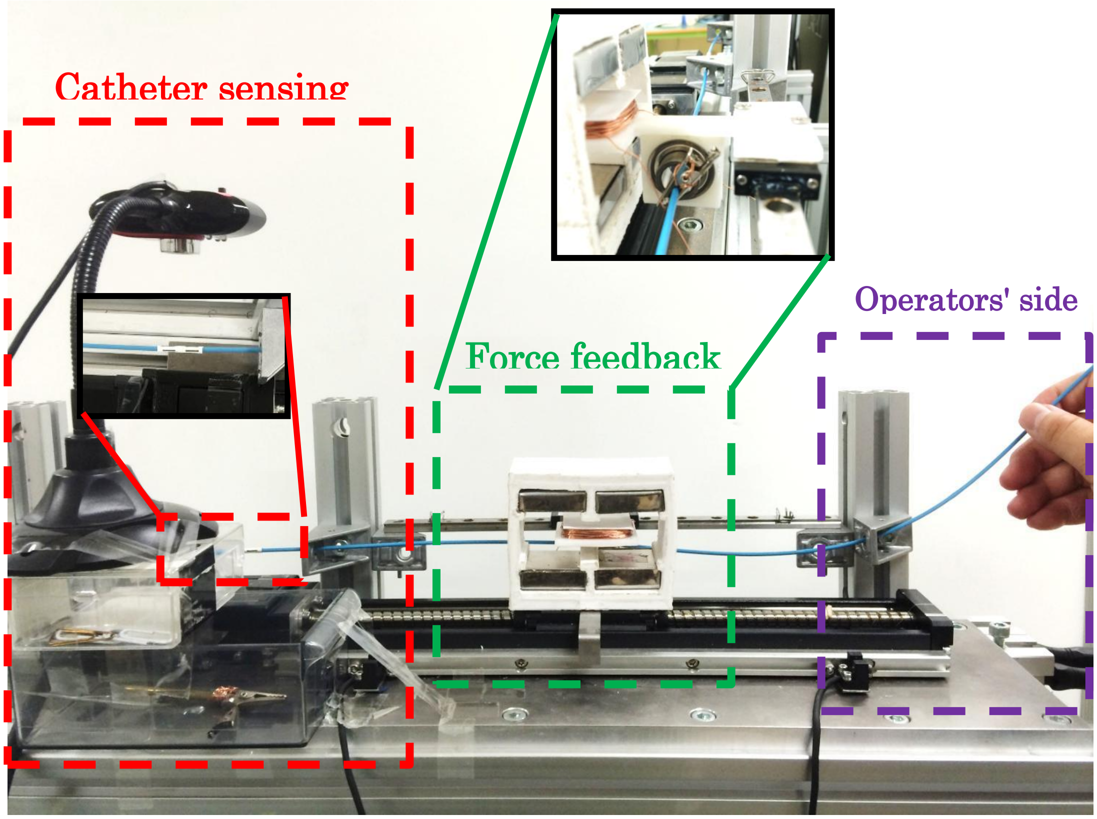
Chapter 5

A new haptic controller for training in Vascular Interventional Surgery

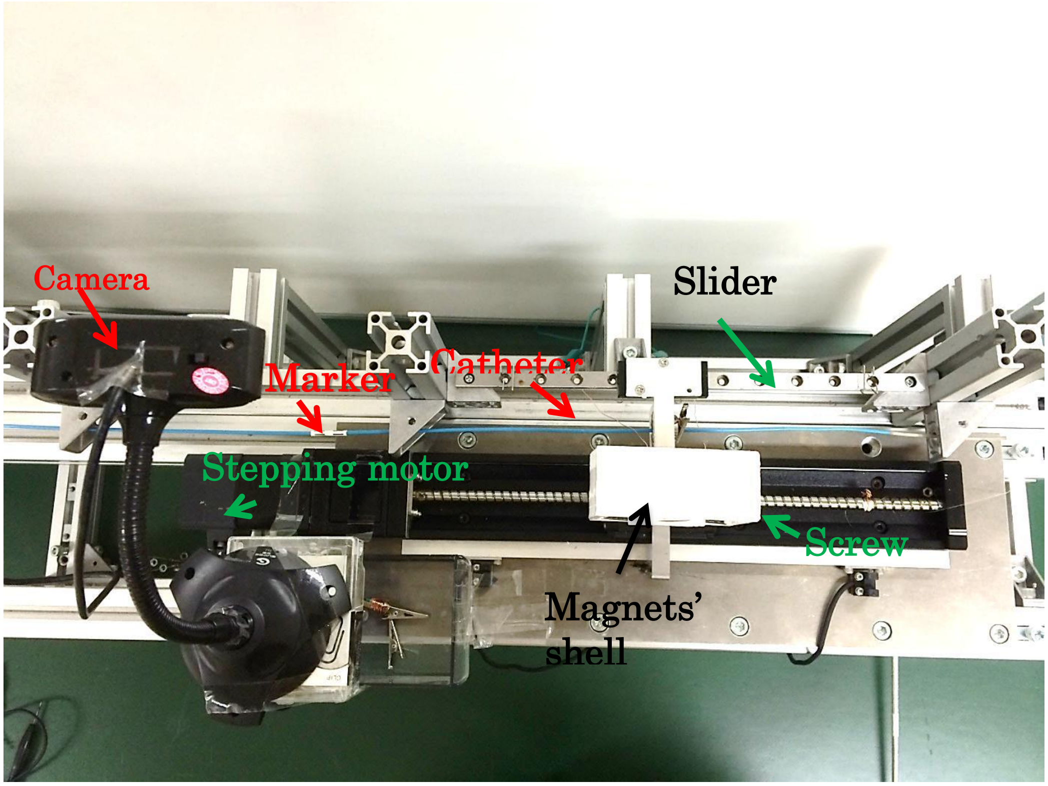
It is difficult to manipulate a flexible catheter to target a position within the patient's complicated and delicate vessels. Training is therefore especially crucial in an endovascular catheterization since any mistake due to the interventionalists' inexperience can be fatal for patients. Robot-assisted and computer-assisted catheterization training have attracted significant interest in recent years. However, few researchers focused on the controller designs with much consideration of the natural catheter manipulation skills obtained from manual catheterization. This chapter presents a new master controller for endovascular catheterization that can allow the interventionalists to apply conventional pull, push and twist of the catheter as used in current practice.

As researchers in [Arata07] have pointed out, the ideal catheterization training systems should be designed with much consideration of the conventional catheter handling skills used in a real endovascular interventional surgery. In this chapter, we focused mainly on the design of a master controller to allow the interventionalists to employ an actual catheter, as the controller, to carry out training courses with force feedback in a ergonomic, compact and simple setting.

5.1 Overview of the proposed master controller



(a) the front view of the master controller



(b) the top view of the master controller

Figure 5- 1 Structure of the proposed new master controller

The proposed master controller consisted of a catheter sensing unit capable of measuring translational and rotational motion of the input catheter, a force feedback unit capable of exerting force feedback on the input catheter, and a control console that transmits information for the force feedback unit and the motors.

A camera was fixed right above the catheter to detect the feature points attached to the catheter. The translational displacement and rotational angles of the input catheter were calculated according to the movements of the feature points. The bearing was fixed in the middle of the holder between the low-friction slider and the coil plate. The catheter passed through a bearing with a grasper which was employed to integrate the catheter with the coil plate. Four permanent magnets were arranged within the magnet’s shell, which was anchored on the movement stage of a screw bearing actuated by a stepping motor. The coil plate was located inside the magnet’s shell, in the middle position between two above magnets and two below magnets. To provide force feedback in axial direction, the magnetic force was generated by the permanent magnets and energized coil. When the interventionalists push or pull the catheter, the stepping motor controls the position of the magnet’s shell keeping the magnet’s shell relatively static to the coils plate according to the displacement of the catheter based on a position-control loop.

5.2 Catheter sensing methods

To avoid additional friction and allow the interventionalists to advance the catheter using the push, pull and rotate technique, a camera was employed to measure the axial and radial motion of the input catheter simultaneously.

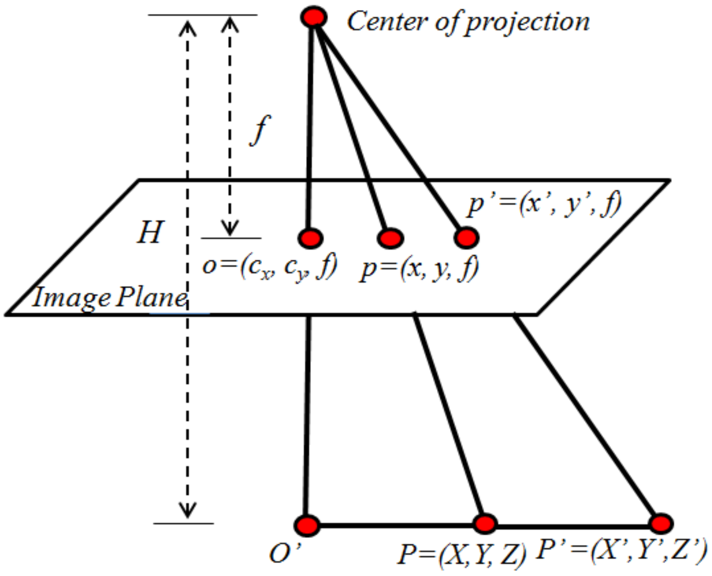
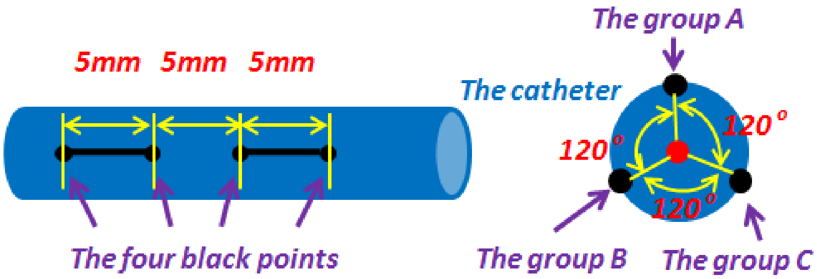
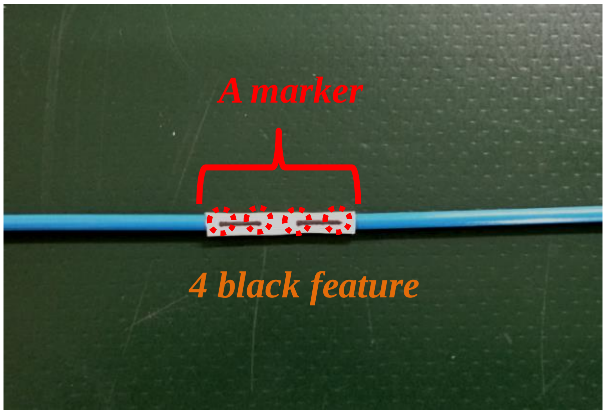


Figure 5- 2 The relationship between the pixel units and physical units

To use a camera to determine the catheter's position, an unavoidable problem is the lens distortion resulted from the reasons of manufacturing. In practice, two main distortions, radial distortions generated due to the shape of lens and tangential distortions resulting from the assembly process of the camera as a whole should be considered. To obtain both the model of the camera's geometry and a distortion model of the lens, the process of camera calibration was performed based on Zhang's method. This method is simple and based mainly on different nonparallel views of a planar pattern, such as a chessboard. The frames of a chessboard pattern were captured from a variety of angles and positions and the corners of the black and white squares were detected and used to calculate the exact locations of the corners in sub-pixel accuracy level. A homograph matrix was identified based on a maximum-likelihood algorithm and the camera's intrinsic parameters were separated from the perspective projection matrix under the assumption of no camera distortion. The extrinsic parameters for each image of the chessboard pattern were then determined, to describe the position of the chessboard pattern relative to the camera in each image. After the intrinsic and extrinsic matrices were estimated, the corresponding equations were solved to determine the distortion parameters.



(a) the distribution of a marker



(b) a marker attached to a real catheter

Figure 5- 3 The markers attached to a 2-mm-diameter catheter.

After the process of camera calibration, the center point of the chip, focal length (f) and five distortion parameters were known values for us. Actually two different focal lengths, fx in horizontal direction and fy in vertical direction, were measured respectively but fx was equal to fy in our case. The focal length f was actually the product of the physical focal length of the lens (units of pixels per millimeter) and the size of the individual imager elements (units of millimeters). Distortion parameters were employed to compute a distortion map that was then used to correct the image. The coordinate values of the detected feature points were transformed from their original coordinates to the corresponding undistorted coordinates based on the distortion map. Assuming that point P and P' are two feature points attached to a catheter, the real distance can be measured as DPP'. The two feature points were then detected in the image captured by the camera. The distance in pixels between point p and point p' was marked as dpp'. H can be calculated based on the principle of similar triangles, as (1) shows:

$$\frac{d_{pp'}}{D_{PP'}} = \frac{d_{Op}}{D_{OP}} = \frac{f}{H} \quad => \; H = \frac{fD_{PP'}}{d_{pp'}} \qquad \text{(Eq. 5-1)}$$

$D_{PP'}$ and f were measurable values. The distance $d_{pp'}$ was computed using $\sqrt{(x' - x)^2 + (y' - y)^2}$. D_{OP} and d_{Op} denote the lengths of line Op and line OP, respectively.

After the calculation for the height H, the relationship between pixel units and millimeter units was built. A distance in image coordinate system (d_{pixel}) can be transformed to a physical distance (d_{mm}) in millimeter, as (2) shows:

$$\frac{f}{H} = \frac{d_{pixel}}{d_{mm}} \quad => \quad d_{mm} = \frac{Hd_{pixel}}{f} \qquad \text{(Eq. 5-2)}$$

To shorten the feature detection time and make it easier compute the translational displacement and rotational angles of the input catheter, several instrument markers were attached to the catheter. We defined the four feature points (black points) as a group and A instrument marker consisted of three groups (group A, B and C). Every two markers deployed had a 8-cm-spacing and every two black feature points deployed in a marker had a 5-mm-spacing. The specific marker distribution is shown in Figure 5-3. Note that the markers were extremely thin, and therefore, did not enlarge the diameter of the real catheter.

To determine the position of the input catheter, we first needed to extract the markers attached to the catheter. Nineteen best corners with large variation in intensity in all the directions were detected based on Shi-Tomasi Corner Detector in a picture captured by the camera. Due to the moving range of the catheter in the camera's view was merely in the middle section, a mask area shown in Figure 5-7(a) was used to specify the region in which the corners were detected. The original image was converted into a grayscale image firstly and we maximized the Eq. 5-3 to find the difference in intensity for a displacement of (u, v) in all directions in the specified region.

$$E(u,v) = \sum_{x,y} \omega(x,y)[I(x+u,y+v) - I(x,y)]^2 \quad \text{(Eq. 5-3)}$$

where $\omega(x,y)$ is a window function and $I(x+u,y+v)$ denotes shifted intensity and $\mathrm{I(x,y)}$ presents intensity. Applying Taylor Expansion to above equation, the final equation was defined:

$$E\ (u,v) \approx [u\ \ v]\, M \begin{bmatrix} u \\ v \end{bmatrix} \quad \text{(Eq. 5-4)}$$

where $M = \sum_{x,y} \begin{bmatrix} I_x I_x & I_x I_y \\ I_x I_y & I_y I_y \end{bmatrix}$ and I_x and I_y are image derivatives in x and y directions, respectively. The scoring function was given by:

$$R = min(\lambda_1, \lambda_2) \quad \text{(Eq. 5-5)}$$

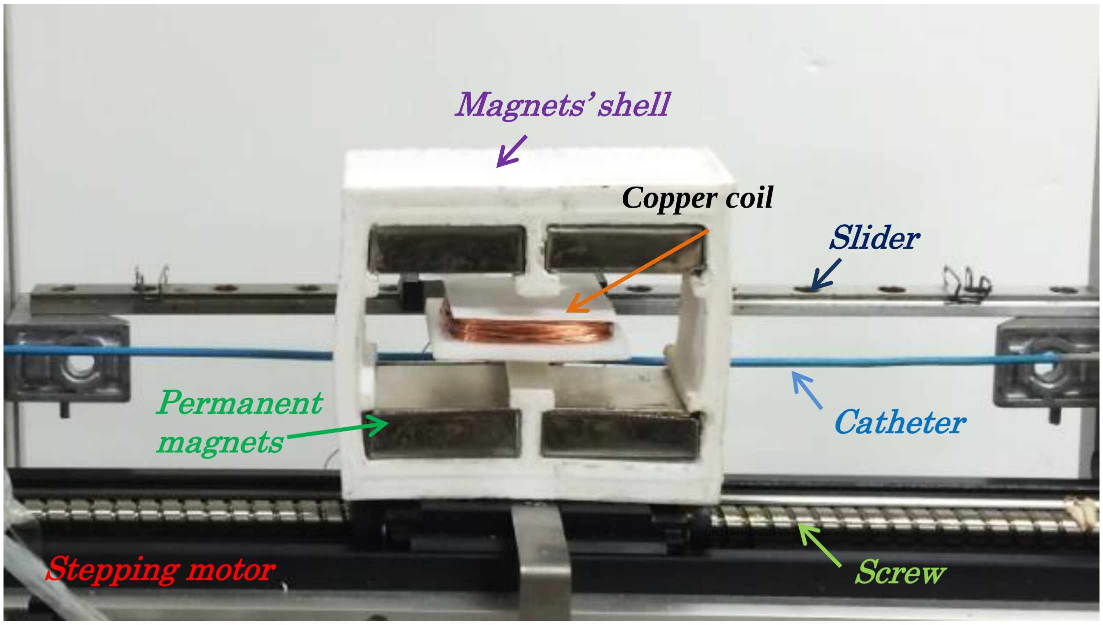
where λ_1 and λ_2 are the eigen values of M. If it is a greater than a threshold value, it is considered as a feature point. This procedure can find the corner points in pixel accuracy.

To achieve better accuracy, sub-pixel level positions of the feature points were computed by interpolating the brightness intensity between the pixels. Firstly, the feature points in pixel accuracy were detected according to the equation above. A small window surrounding the detected position of the feature point was then introduced to define the brightness intensity values between pixels using interpolation. We defined step 0.1 pixel between original pixels in which case 1 pixel contains 10 sub-pixels. We can therefore achieve 10 times higher accuracy than pixel level detection. For each feature point q in pixel accuracy, every vector from point q to a point p located surrounding q was orthogonal to the image gradient at p subject to measurement and image noise. The sum of the errors S for point q was defined as (Eq. 5-6) shows:

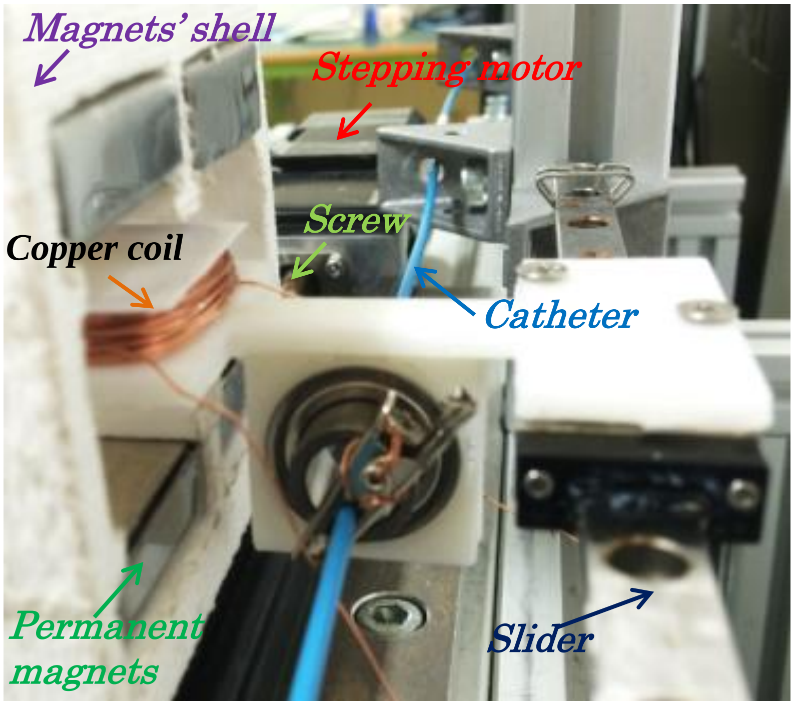
$$S = \sum_i \nabla H_i^T \cdot (q - p_i) \qquad \text{(Eq. 5-6)}$$

where ∇H_i^T is an image gradient at one of the points p_i in a neighborhood of q and the iterative algorithm was employed to find the sub-pixel position where S had the minimum value.

After applying the feature detection algorithm, nineteen strongest corners in the specified region were captured, in which four black points in a marker were the desired feature points and the other fifteen points were defined as noise that should be removed. We achieved this purpose based on the distribution rules of the four black points that every two black points deployed in a marker had a 5-mm-spacing and four black points in a marker had almost equal coordinate values in y direction. The detection results are shown in Figure 5-7 and Figure 5-8.



(a) the front view of the force feedback unit



(b) the lateral view of the force feedback unit

Figure 5- 4 The design of the force feedback unit

The origin of the image was defined at the upper left part. The average values of the difference of the four black points between the previous frame and the current frame in x and y direction were calculated as the horizontal and vertical displacement of a marker respectively, as Eq. 5-7 shows:

$$d_x^{pixel} = \left(\sum_{i=1}^{4}\left(X_{cur}^{i} - X_{pre}^{i}\right)\right)/4$$

$$d_y^{pixel} = \left(\sum_{i=1}^{4}\left(Y_{cur}^{i} - Y_{pre}^{i}\right)\right)/4 \qquad \text{(Eq. 5-7)}$$

where X_{cur}^{i} and X_{cur}^{i} present the pixel coordinates of a black point *i* in x direction in the current frame and in the previous frame respectively. Y_{cur}^{i} and Y_{cur}^{i} present the pixel coordinates of a black point *i* in y direction in the current frame and in the previous frame respectively. Additionally, the translational displacement (d_x^{mm}) and rotational angles (θ°) of a catheter were computed using the following according to Eq. 5-8 and Eq. 5-9.

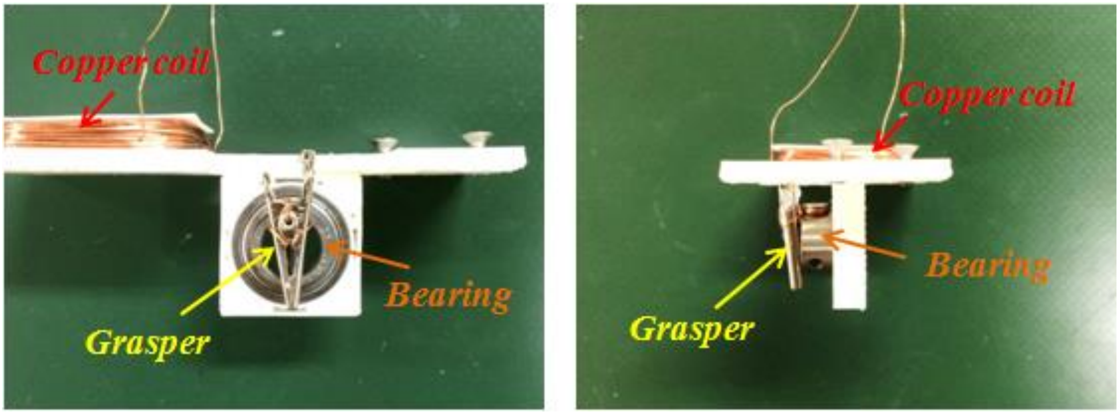
$$d_x^{mm} = d_x^{pixel} \cdot \frac{f}{H} \qquad \text{(Eq. 5-8)}$$

$$\theta^{\circ} = \frac{180 \cdot f}{H \cdot \pi \cdot r} \cdot \sigma \cdot d_y^{pixel} \qquad \text{(Eq. 5-9)}$$

where *r* is the radius of the catheter and σ is a compensation coefficient for the calculation of rotational angles. It is unavoidable to lead to accuracy loss using a line segment ($d_y^{pixel} \cdot \frac{f}{H}$) to approximate the arc length. I therefore introduced a compensation coefficientσto make the approximate values close to the true values. σ was defined as the quotient of arc length divided by the corresponding line segment in 120^{o}.

5.3 Design of the force feedback unit

Compared to an open surgery, endovascular catheterization techniques have unavoidably reduced the sense of touch. Interventionalists have to depend more on the force feedback resulting from catheter-tissue interactions. Although the existing vascular imaging techniques can provide interventionalists with real-time visual feedback about the interactions between the catheter and the patient's vascular structure, haptic feeling is still an important part to the success of the endovascular interventional procedure.



(a) a front view (b) a lateral view

Figure 5- 5 The design of the bearing with a grasper

The proposed force feedback unit (shown in Figure 5-4) consists of a copper coil plate that was fixed on a holder connected to a slider, a shell with four permanent magnets within it anchored on the movement stage of a screw bearing actuated by a stepping motor, and a bearing with a grasper.

A bearing with a grasper (shown in Figure 5-5) was employed to clamp the catheter. The catheter passed through the bearing and was integrated with the holder by the grasper. The purpose of the design is noninterference between the two degrees of freedom. Namely, the design ensures that the axial movements and force feedback do not affect the radial motions and torque feedback.

A hand-wound copper coil was sandwiched between two sheets of plastic board. The coil plate was incorporated with the catheter by the grasper as the actuator that provides force feedback to the interventionalists. The plastic magnets' shell, produced by a 3D printer, anchored two permanent magnets above the coil. In the same way, the other two matching permanent magnets were anchored below the coil plate. An exploded view of the coil plate and its permanent magnet assemblies is shown in Figure 5-4. When the prototype was assembled, the distance between the upper and lower magnets was set to about 20 mm with the 10 mm thick coil plate situated in the middle of the gap.

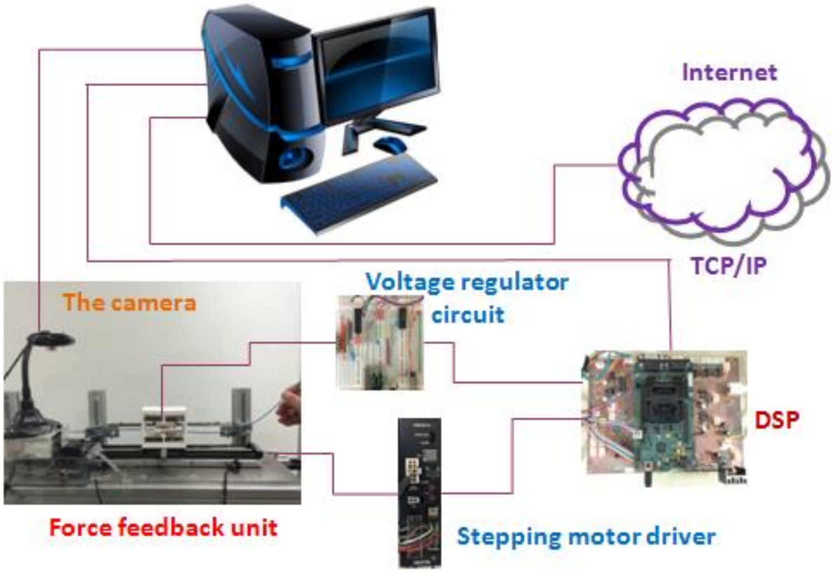
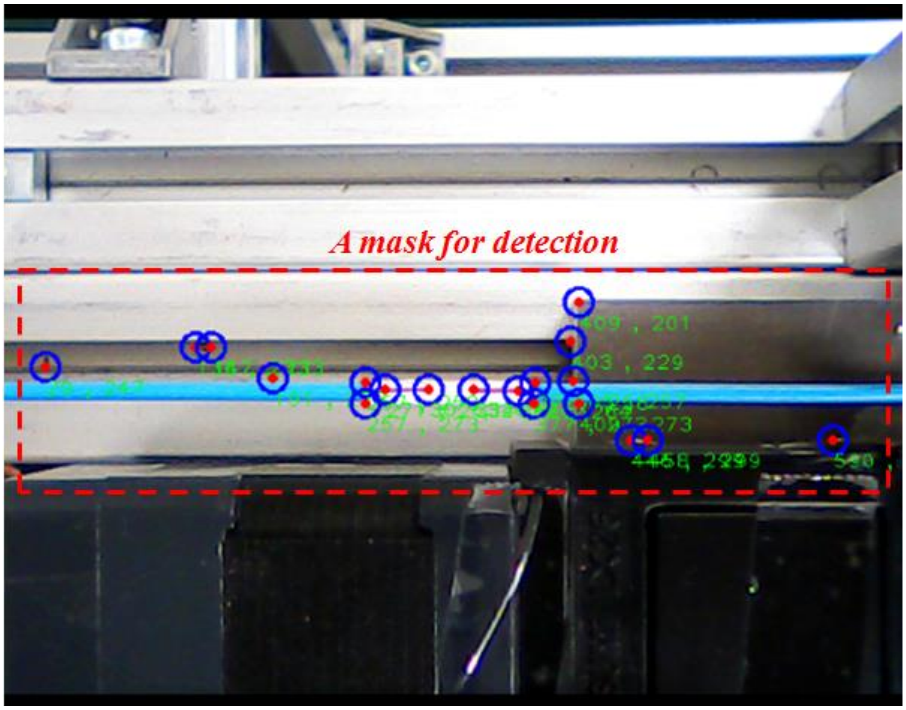
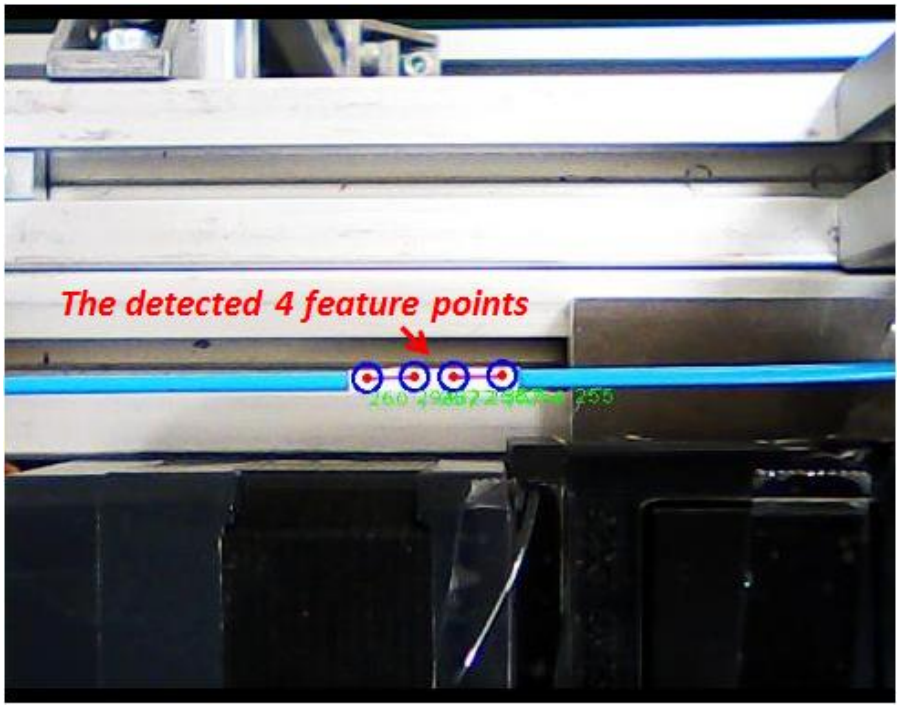


Figure 5- 6 The schematic of the control system



(a) the strongest corners detection



(b) 4-feature-point detection

Figure 5- 7 The results of the feature detection algorithm

When the coil was energized by a coil current I, it can generate a force between the powered coil and the magnetic field B given by (Eq. 5-10):

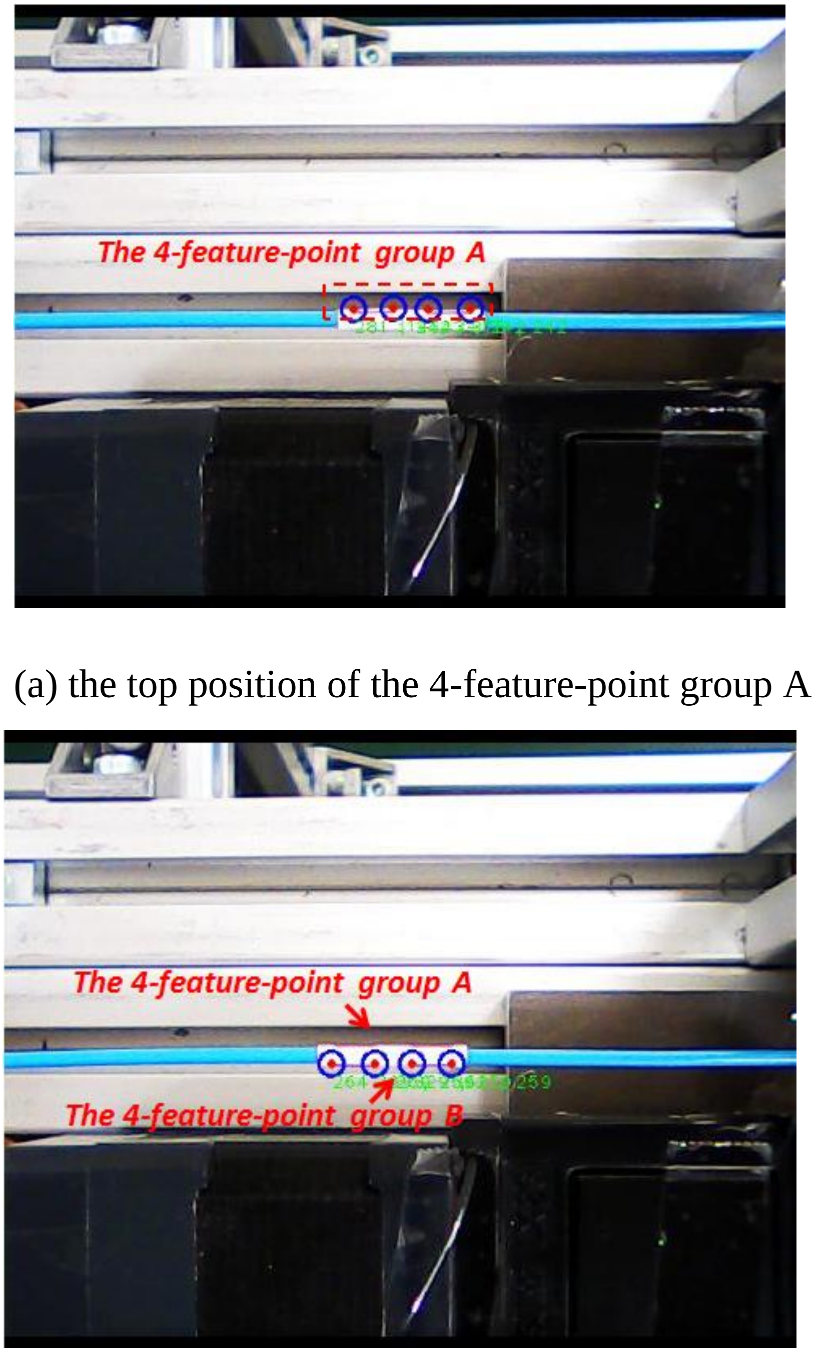
$$F = -I \int_{C} B \times dl \qquad \text{(Eq. 5-10)}$$

where dl is a differential element of wire in the coil, and the integration was calculated over the whole wire between the two pairs of the magnets. The four permanent magnets have powerful magnetic field intensity in our case. Thus the magnetic field B is approximately constant between the two pairs of magnets. The generated force by the interaction between the electrified coil and the magnetic field acted linearly along the slider with its orientation decided by the direction of the current. The measured resistance of the coil was 2 Ω. The force resolution was approximately 25mN and a maximum current of 3A generated an approximately 1.5 N force.

Additionally, lubricants were used on the catheter holders to decrease frictional resistance. Thus we ignored the friction between them and mostly focused on the slider's friction. A loadcell (TU-UJ, TEAC Corp., Japan) fixed on the movement stage of a screw bearing actuated by a stepping motor was introduced to measure the slider's friction. The loadcell can contact the slider and the movement stage was controlled to push and pull the slider to measure the friction. When there is no force feedback to the haptic unit, the coil was powered by the DSP controller to produce a initial force that has the same magnitude but in the opposite direction with the slider's friction so that the pull and push feels frictionless.

5.4 Control system mechanisms

Control of the new master controller was achieved by a computer console and a DSP unit (shown in Figure 5-6). The DSP unit communicated with the computer console via RS-232 serial port.


The 4-feature-point group A
The 4-feature-point group A
The 4-feature-point group B

(a) the top position of the 4-feature-point group A

(b) the bottom position of the 4-feature-point group B

Figure 5- 8 The process of capturing feature points for the rotational calculation.

Control software was implemented in C++ to enable simultaneous motion measurement in the axial and radial directions. The axial position of the input catheter was transmitted to the DSP unit through RS-232 serial port. To prolong the operation range, the magnet's shell followed the motion of the coil plate to provide the magnetic field. A PID control loop implemented on the DSP unit was employed to control the relative position between the magnets' shell and the coil plate running at 1 kHz. Namely, the PID control loop aims to maintain a zero offset between the movement stage and the catheter. The encoder measured the position of the movement stage so that the difference between translational displacement of the catheter and the measured displacement forms the input to the PID controller. To output a stable voltage to the coil, a voltage regulator circuit was designed to control the current of the force feedback device by the DSP unit running at 1 kHz.

Additionally, the control software implemented on the computer enable to control the simulated catheter in virtual reality training systems or to drive a slave manipulator in the axial and radial motion based on the Internet.

5.5 Experimental results

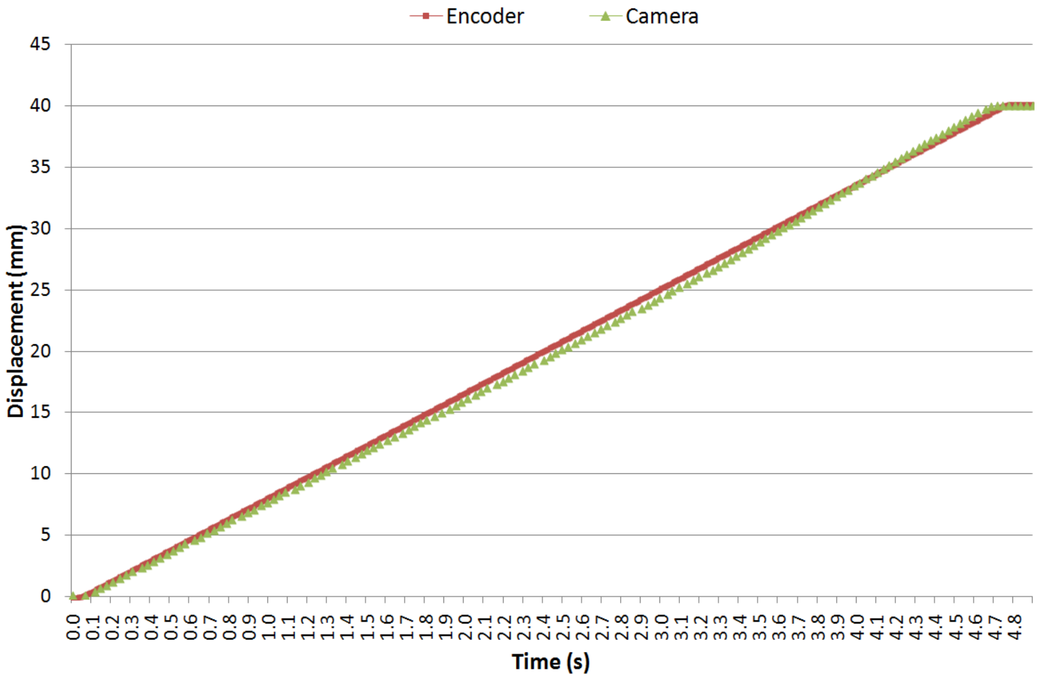
To demonstrate the performance of catheter sensing unit and the force feedback unit, the proposed master controller was evaluated with a series of experiments.

5.5.1 Evaluation of feature extraction algorithm

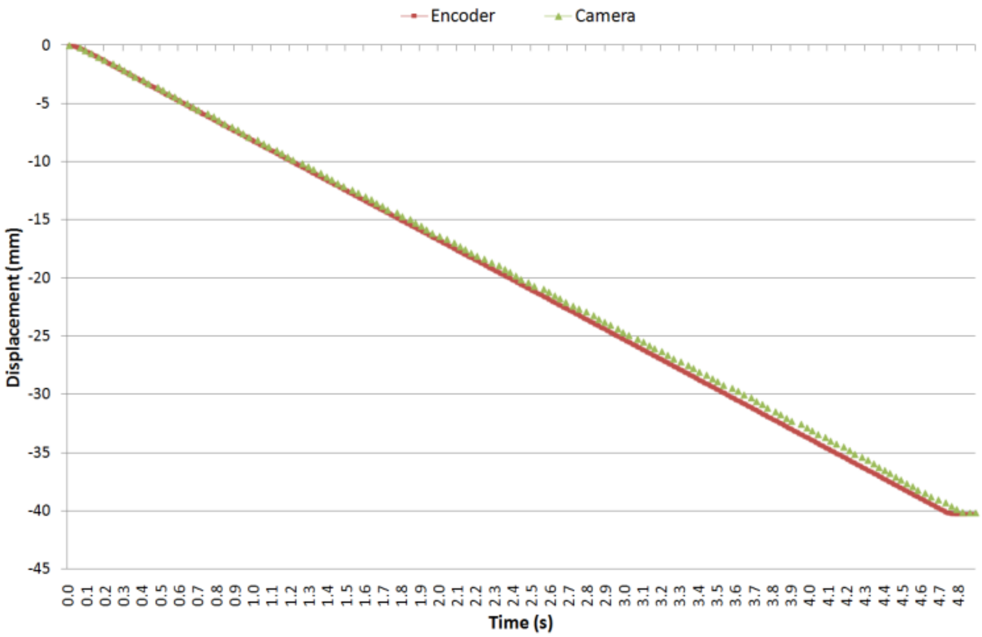
According to the feature extraction algorithm described above, the results of the method are shown in Figure 5-7 and Figure 5-8. I first detected the nineteen corner points with large variation in intensity in the mask region. The noise points were then removed based on the distribution rules of the four black points. The average displacement of the four feature points of group A, group B or group C in x direction was used to determine the axial motion of the catheter. For the radial motion, the average displacement of the four feature points of group A in y direction was applied to compute the rotational angles of the catheter. When the group A reached its travel limit, group B was then detected to measure the radial motion of the catheter instead of group A. The measurement range of the rotational angles for each group was 120°.

5.5.2 Evaluation of catheter sensing unit

To evaluate the performance of the catheter sensing unit, a stepping motor with a encoder was employed to pull, push and rotate the catheter and compared the data obtained by the encoder with translational and rotational results computed by the catheter sensing unit. The insertion and extraction were conducted at an approximately constant velocity of 8.4 mm/s. The rotation was performed at an approximately constant angular velocity of 115 °/s. The data obtained from encoder was used as the benchmark. The comparative results of insertion and extraction motion and rotation motion are shown in Figure 5-9 and Figure 5-10 respectively.

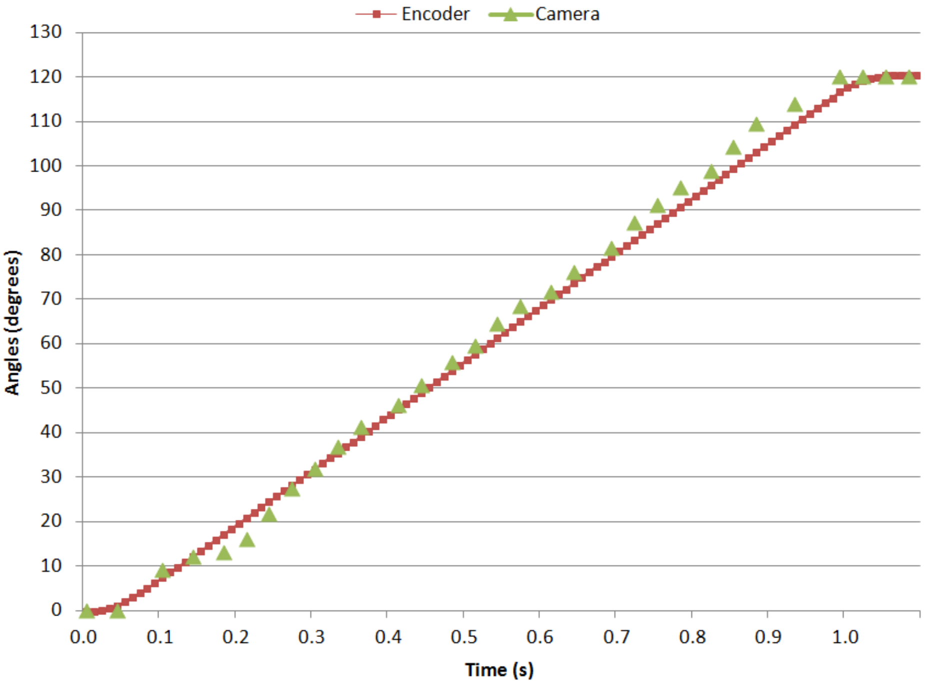


(a) the insertion motion

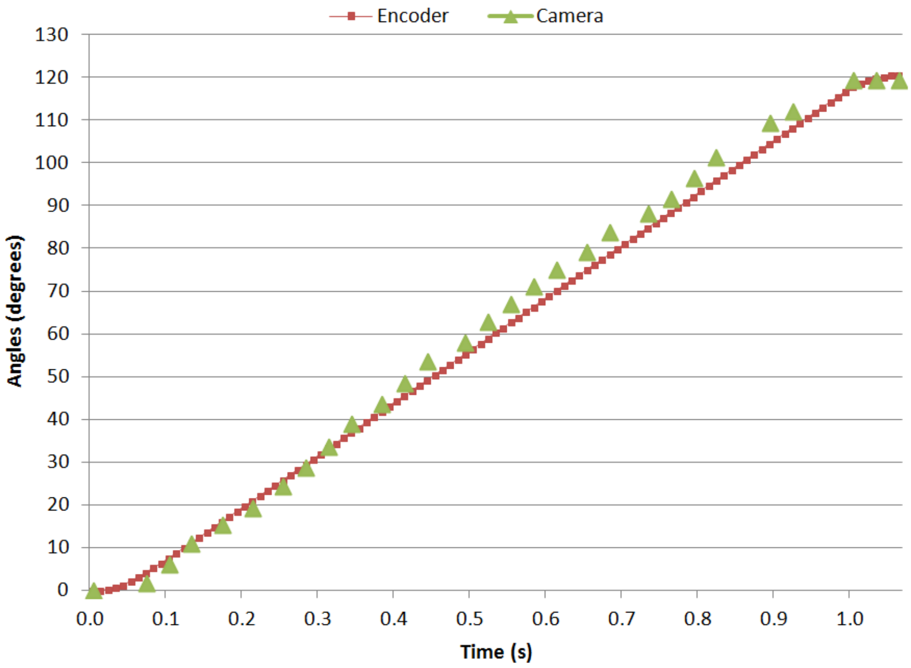


(b) the extraction motion

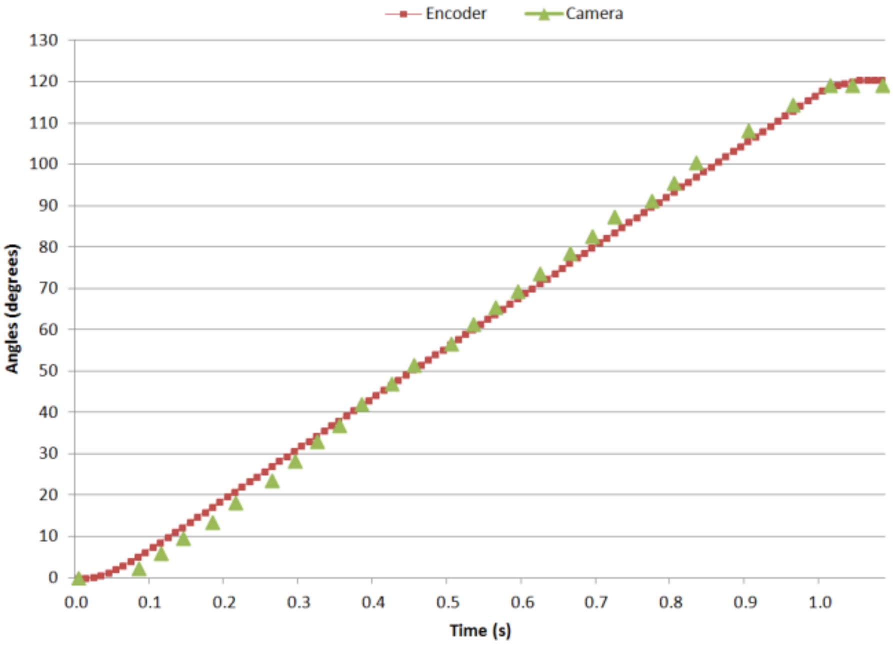
Figure 5- 9 the results of the insertion and extraction motion



(a) the 4-feature-point group A



(b) the 4-feature-point group B



(c) the 4-feature-point group C

Figure 5- 10 The results of the rotation motion

Additionally, another 20 groups of insertion and extraction experiments were performed. The results of insertion/extraction experiments with the same displacement (40mm) at different velocities are shown in Figure 5-11 and Figure 5-12 presented the comparative results with the same velocity (33.6mm/s) in different displacements. Furthermore, the other experimental results for radial motion of the catheter between the encoder and the catheter sensing unit are shown in Figure 5-13. The measurement range of each marker for rotational angles was 0 º~120 º. From the experimental results, the minimum and the maximum errors of translational displacement were 0.003 mm and 0.425 mm respectively. The average error was 0.113mm. In terms of rotational angles, the minimum and the maximum error was 0.39 º and 7.2 ºrespectively. The average error was 3.61 º.

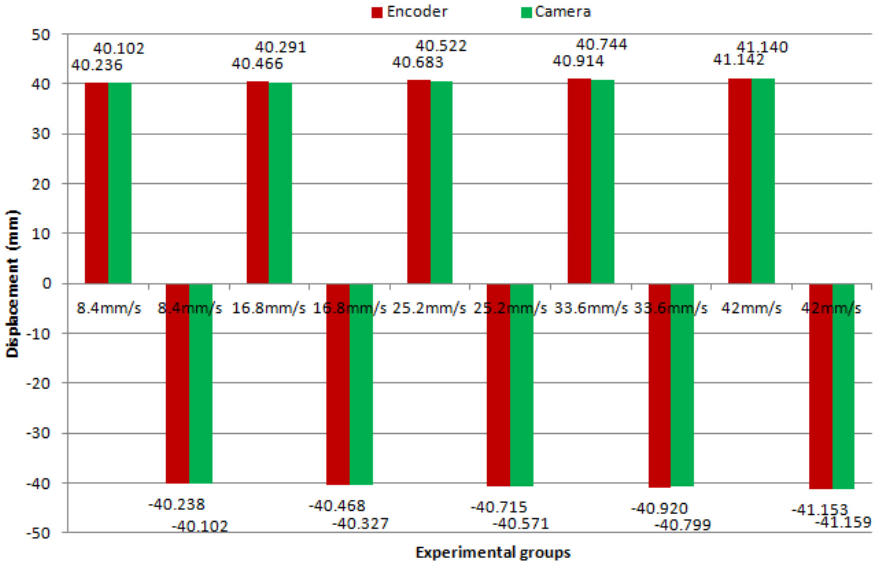


Figure 5- 11 The results of pull/push experiments with different velocities

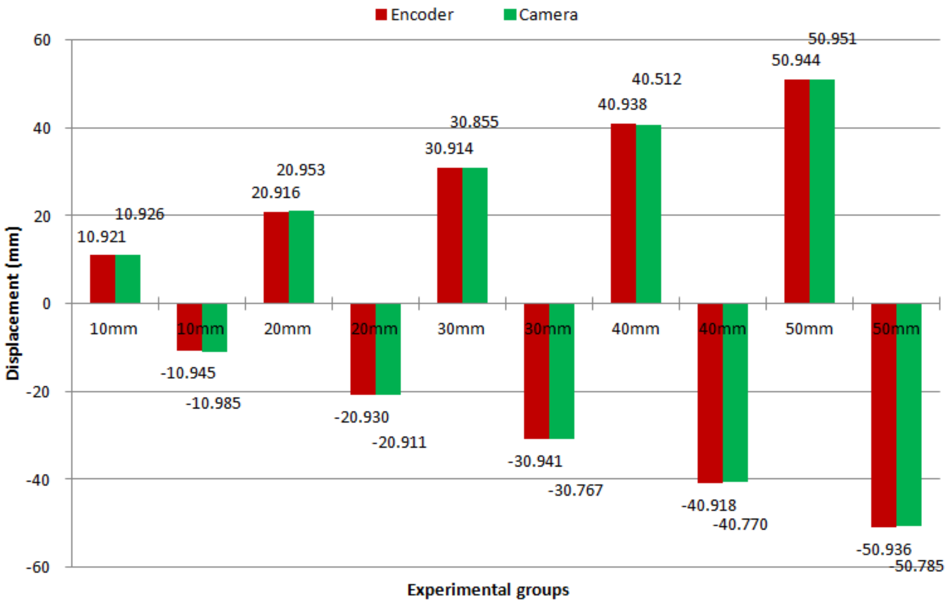


Figure 5- 12 The results of pull/push experiments with different displacement.

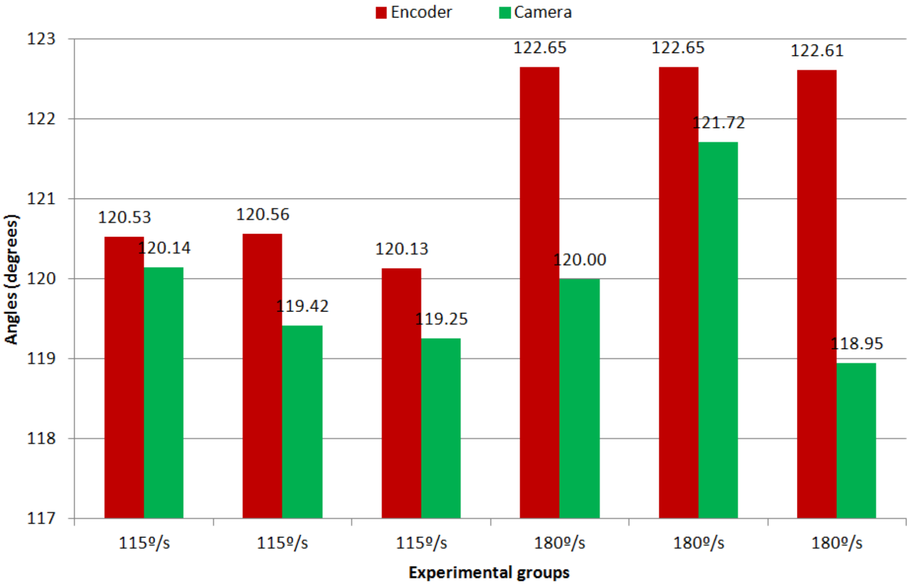


Figure 5- 13 The results of the rotation experiments with different velocities.

5.5.3 Evaluation of force feedback unit

To provide the force feedback based on the magnetic force generated by the permanent magnets and the powered coil, the first step is to build a relationship between the current and the magnetic force. A loadcell was used (shown in Figure 5-14) to measure the force generated by the force feedback unit. A regulated DC power supply was then introduced to provide the excitation current to the coil gradually increasing from 0 to 1.08 A by step 0.01 A. A trend line was defined as the relationship between current and the force which is shown in Figure 5-15. It was almost a linear relationship with the fitted linear equation given by (Eq. 5-11):

$$y = 0.5817x - 0.071 \qquad \text{(Eq.5-11)}$$

where y presents the force and x is the input current. Due to the slider's friction, the initial current was set to 0.14 A.

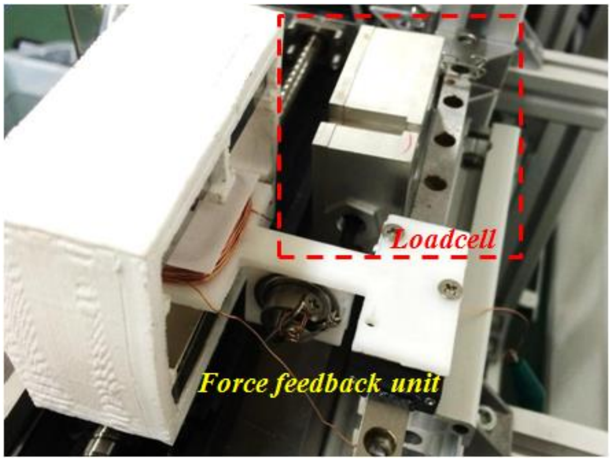


Figure 5- 14 The experimental setup for the evaluation of the force feedback unit

When the force measured by the slave system back to the master controller, the current can be computed according to the force based on (Eq. 5-11). To evaluate the performance of the force feedback unit, we assume that the measured force by the slave system like the blue curve in Figure 5-16 back to the master controller. The force feedback unit controlled the current to generate the corresponding force based on relationship between the current and the magnetic force. Additionally, a loadcell was employed to measure the force generated by the powered coil and the magnetic field. The results are shown in the red curve in Figure 5-16. The minimum error was 0.0052 N and the maximum error was 0.0247 N.

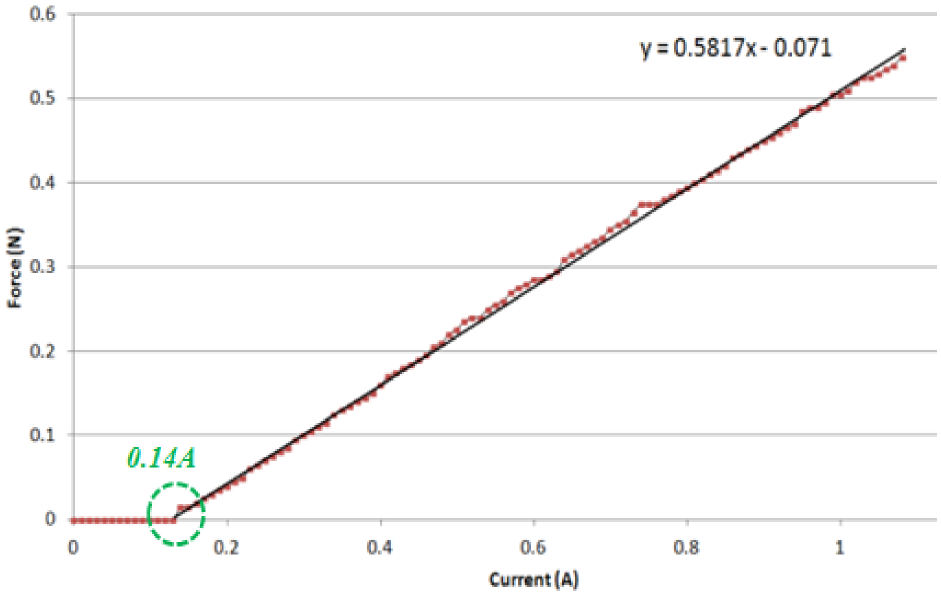


Figure 5- 15 The calibration results between the current and the force

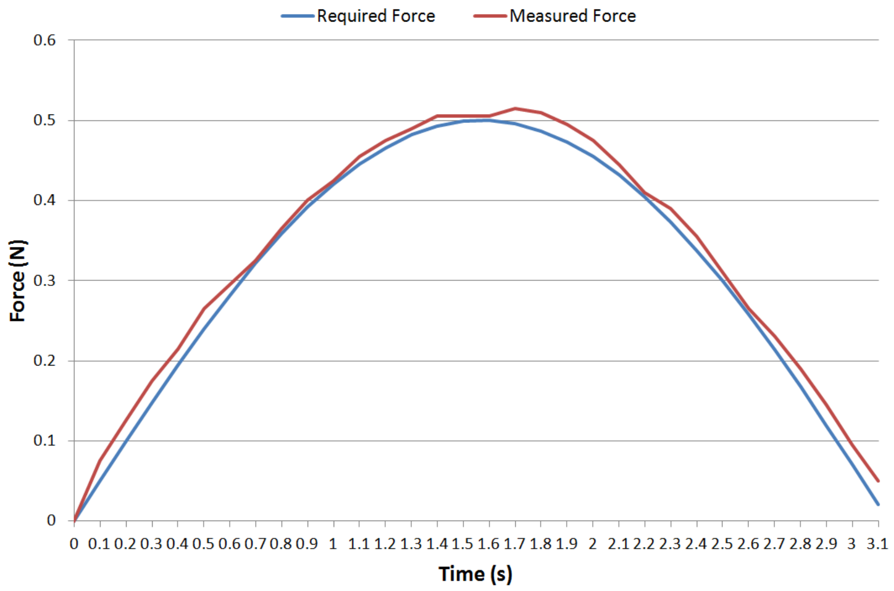


Figure 5- 16 The comparative results between the require force and the measured force

The new master controller described in this paper was designed to operate in two modes. It can be employed as a controller to enable interventionalists to use their highly developed dexterous skills to remotely maneuver the slave catheter manipulator, potentially reducing radiation exposure and physical stress during long procedures. The slave system can connect to the master controller based on Internet for remote experiments over inter-country distances or the cable for short distance operations. The resultant force at the proximal end of the catheter resulting from catheter-tissue interactions sensed by the slave system can exert on the catheter of the new master controller to provide operators with a sense of resistance. Also, it can be used to pull, push or twist the simulated catheter in virtual reality-based training systems.

The VR-based simulators offer flexible scenarios with a high degree of realism. Interventionalists are allowed to build the desired scenario according to the corresponding CT (Computerized Tomography) or MRI (Magnetic Resonance Imaging) dataset. Additionally, the elastic characteristic can be set to simulate different kinds of soft tissues. The interaction force between the simulated catheter and its surrounding virtual vascular structure can be provided to the operators' hands via the force feedback unit of the proposed master controller.

Compared to the previous designs, the proposed master controller allowed a contactless measurement which avoids additional friction and provided contactless force feedback in the axial direction to the interventionalists. Moreover, the design enforced the catheter straight below the camera regardless of how the interventionalists push, pull or twist the catheter. The interventionalists can manipulate the catheter according to their own habits. It is applicable not only to catheterization but also to similar interventions. The current implementation of the master controller was designed for use with 6-7 F catheters and the design was easily adaptable for catheters of larger sizes. Utilizing this design with smaller catheters will require modifying the markers attached to the catheter. A short hollow cylinder with larger size can be used to load the markers below the catheter sensing unit. The catheter then passes through the hollow cylinder and can rotate coaxially with the hollow cylinder. The marking method and the image processing algorithms are still the same with the proposed catheter sensing unit.

Additionally, the proposed master controller has many potential advantages over commercially available systems. Unlike most commercial catheterization training systems utilizing joystick-type input devices or specialized catheters to navigate the slave manipulator or simulated catheter in VR system, the proposed master controller has the ability to employ actual and generic catheters. It also allowed to apply conventional push, pull and twist of the catheter during the catheterization procedure. Additionally, the proposed master controller has a simple and compact structure and is more cost-effective.

5.7 Summary

Based on analysis of requirements and state of the art in computer-assisted and robot-assisted training systems for endovascular catheterization, a new master controller with force feedback interface was proposed to maintain the natural endovascular catheterization skills of the interventionalists. A camera was employed to detect the axial and radial motion of the catheter simultaneously with the ability to allow interventionalists to use their dexterous skills while performing catheter-based interventions. Additionally, a force feedback unit was developed to provide the operators with a sense of resistance force. A series of the experiments were carried out to evaluate the performance of the catheter sensing unit and the force feedback unit.

Future work includes the development of virtual reality-based training system based on the proposed master controller and a user study to demonstrate the effectiveness of our system.

Chapter 6
The 2D and 3D models for reducing the time delay of visual feedback based on virtual reality techniques

6.1 2D Model for reducing the time delay of visual feedback

6.1.1 The Implementation Method at the slave side

At the slave site, a fire-wire digital camera capturing images at 30fps has been used to obtain images as the catheter is inserted into the test-bed. The images obtained from this camera are similar to X-ray fluoroscopic images in terms of contrast and frame capture rate. However, X-ray fluoroscopic images have a much higher resolution as compared to the images obtained from the fire-wire camera. At the same time, the images are processed using a novel real-time algorithm to track the tip of the catheter to get the coordinate values in the Cartesian coordinates and we set the top-left point as the original point. Then the coordinates are sent to the master side. The size of coordinates is much less than that of image information. The real-time algorithm of motion tracking as showed in Figure 6-1 below.

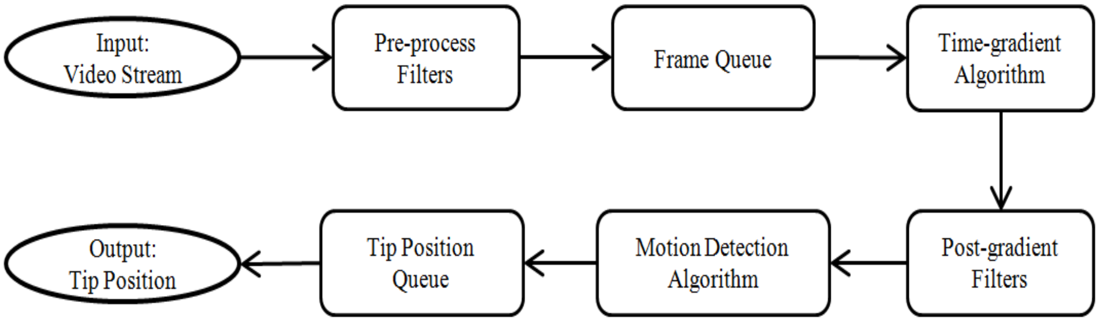


Figure 6- 1 The structure of the surgeon console

The video stream consists of a stream of frames grabbed by the camera at 30fps. All frames are time-stamped. The pre-process filter block consists of a set of filters for suppressing noise, masking the image and improving the contrast of the image by adaptive thresholding. The frame queue is an image buffer structure managed by the queue manager block. This queue provides the appropriate input for the time-gradient algorithm. The time-gradient algorithm takes time gradients of the images in the frame queue. As the insertion speed is finite and limited, it concentrates on a neighborhood of the previous valid catheter tip position to limit the search area and decrease the processing time. The post-gradient filters suppress the noise in the gradient image and make it useful for the motion-detection algorithm. The motion-detection algorithm extracts the linear speed vector of the catheter (along the catheter axis) as well as the tip position. It also generates a true/false flag called motion flag which is true when any motion is detected. The signals generated by the motion detection algorithm are fed back to the master side.

6.1.2 The Implementation Method at the master side

At the master side, we used canny edge detection algorithm to get the navigation chart which is very similar to the blood vessel model at the slave side. The position of the tip of catheter are reconstructed in the navigation chart according to received coordinate values from the slave site. The navigation chart is also defined in Cartesian coordinates. The canny edge detection algorithm was developed by John F. Canny as a means to detect edge lines and gradients for the purpose of image processing. This algorithm provides good detection and localization of real edges while providing minimal response in low noise environments. This algorithm is well known and explained in any introductory text on image processing. The main stages of the Canny Algorithm are as follows: Noise reduction by filtering with a Gaussian blurring filter; Determining the gradients of an image to highlight regions with high spatial derivatives; Relate the edge gradients to directions that can be traced; Tracing valid edges; and Hysteresis thresholding to eliminate breaking up of edge contours.

6.1.3 Experimental results

To testify the accuracy of our method, we operated the handle of master manipulator to insert or rotate catheter, throughout the aortas located in the slave side of the master-slave catheter operating system, as showed in Figure 6-2, using Endo Vascular Evaluator Model (EVE) to simulate blood vessels and load cell. EVE model is made of a special silicone that recreates the elasticity and friction of human vasculature, simulating the sensation and behavior of catheter manipulation. The bending angles and radii of the tubes and the elastic and damper coefficient in the EVE are close to those of human arteries. So it can be used for blood vessel simulation.

At the same time, the controlling instructions are transmitted to the slave side. The controller in slave side can insert or rotate catheter according to the controlling instructions from master side. Meanwhile, the motion tracking program running in the slave side capture the coordinate values of the tip of the catheter and transmit the coordinates to the master side. Figure 6-3 shows the coordinates of the tip of the catheter in the slave side. These coordinates are used for the correction of the tip of the catheter in the master side. Figure 6-4 shows the path in the navigation chart according to coordinates received from slave side.

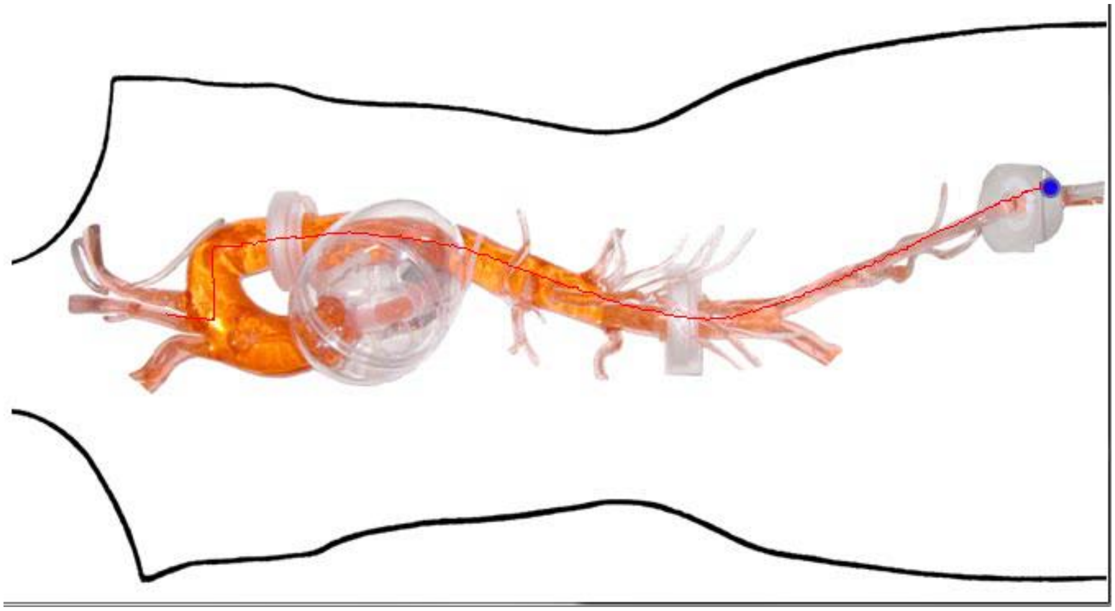
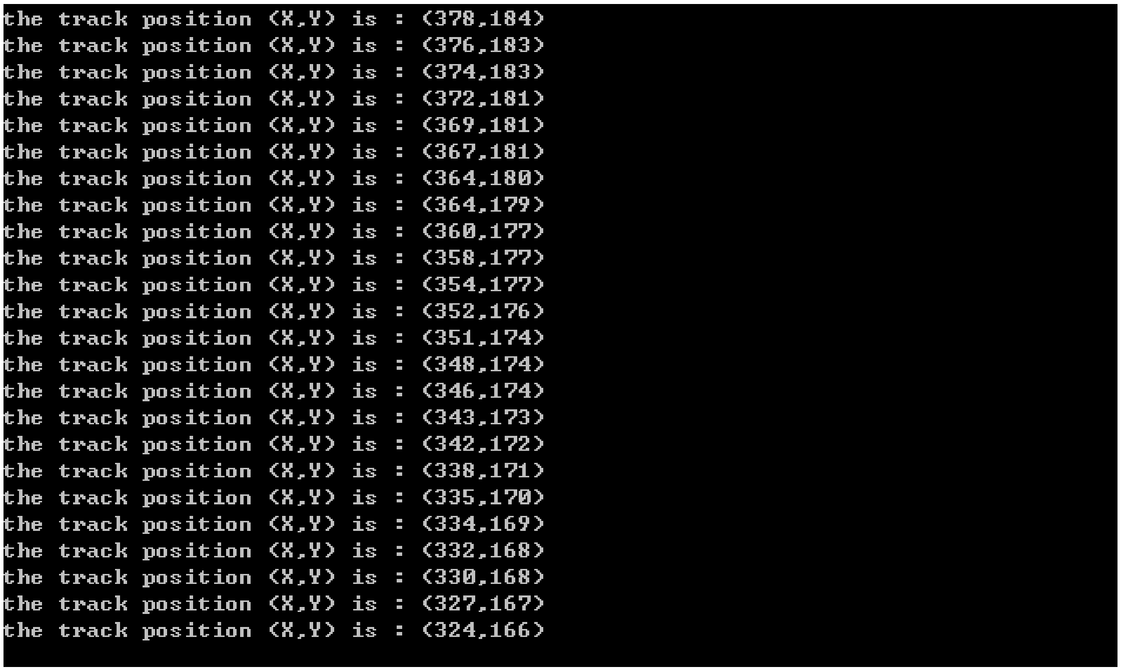


Figure 6- 2 The path of the tip of the catheter in the slave side



```
the track position (X,Y) is : (378,184)
the track position (X,Y) is : (376,183)
the track position (X,Y) is : (374,183)
the track position (X,Y) is : (372,181)
the track position (X,Y) is : (369,181)
the track position (X,Y) is : (367,181)
the track position (X,Y) is : (364,180)
the track position (X,Y) is : (364,179)
the track position (X,Y) is : (360,177)
the track position (X,Y) is : (358,177)
the track position (X,Y) is : (354,177)
the track position (X,Y) is : (352,176)
the track position (X,Y) is : (351,174)
the track position (X,Y) is : (348,174)
the track position (X,Y) is : (346,174)
the track position (X,Y) is : (343,173)
the track position (X,Y) is : (342,172)
the track position (X,Y) is : (338,171)
the track position (X,Y) is : (335,170)
the track position (X,Y) is : (334,169)
the track position (X,Y) is : (332,168)
the track position (X,Y) is : (330,168)
the track position (X,Y) is : (327,167)
the track position (X,Y) is : (324,166)
```

Figure 6- 3 The coordinates of the tip of the catheter in the slave side

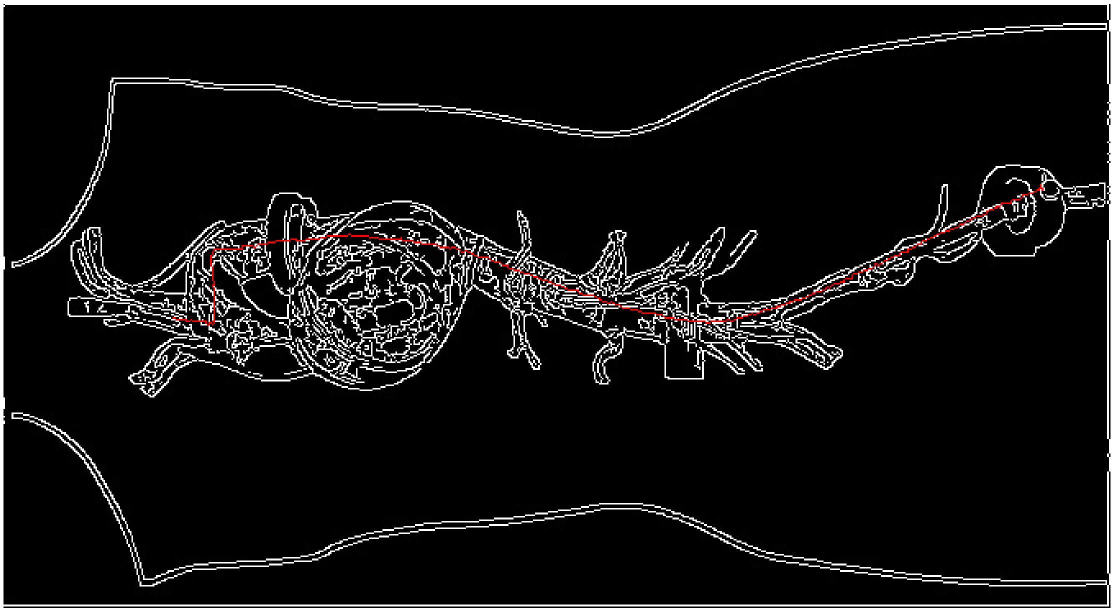


Figure 6- 4 The reconstructed path of the tip of catheter in the master side

We can see that the path in the master side is similar to the actual path in the slave side. And the information needed to transmit is just the coordinates which is much less than image information.

The objective of this work was to develop a reliable way to reduce the time delay of image transmission and the time difference between force signals and image information. In this paper, we designed and implemented a new method that transmitting the visual information to the coordinates values. In the slave side, the coordinates are obtained in real-time, and recovered in the navigation chart in the master side. Therefore, doctors can see the position of the catheter in real-time. Through our method, the size of image information is almost equal to force signals and controlling instruction.

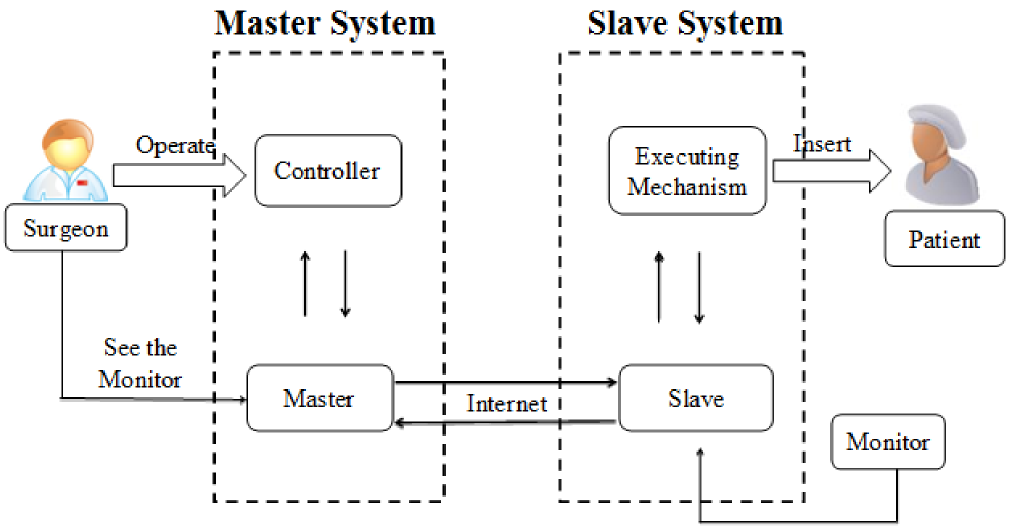


Figure 6- 5 The general structure of Internet-based tele-surgical systems

6.2 3D models for reducing the time delay of visual feedback based on virtual reality techniques

As many researchers have pointed out, time delay in the transmission of visual feedback from the patient's location to the operation location is a critical problem often encountered in tele-operative surgery. This time delay decreases operability by directly interacting with the remote environment. In our proposed robotic catheter operating system, the time delay is defined as follows (Eq. 6-1):

$$T_{total} = T_{transmission1} + T_{response} + T_{tansmission2} \qquad \text{(Eq. 6-1)}$$

$T_{transmission1}$ represents the transmission delay of the control signal from the master systems to slave systems, $T_{response}$ represents the response time of the slave system, and $T_{transmission2}$ is the transmission delay of visual or haptic feedback from the slave system back to the master system. In this paper, we focused mainly on a method of reducing the volume of visual feedback data, thereby decreasing the transmission delay of image information.

6.2.1 Structure of the Proposed Method

Figure 6-5 shows the entire structure of the proposed Internet-based tele-operative surgical system. Two cameras from different viewpoints, one on the left of the catheter and the other on the right, were used to simultaneously capture images of the catheter inside a plastic tube representing the blood vessel at the patient’s location. The two images were then sent to an image processor module to calculate the three-dimensional (3-D) coordinates of the feature markers, which were attached to the catheter. These markers were used to represent the catheter shape. The image information was transmitted as 3-D coordinates via the Internet. At the operation location, based on the 3-D coordinates received from the patient’s location in real time, the shape of the catheter was reconstructed within the corresponding 3-D simulated vessel model, having the same size and shape as the plastic tube. Physicians would then decide whether or not to proceed with catheter insertion/rotation, based on the status of the catheter model inside the simulated vessel in the virtual reality environment. Thus, compared with conventional video signal feedback, this technique provides surgeons with a “live” 3-D virtual representation of the scene.

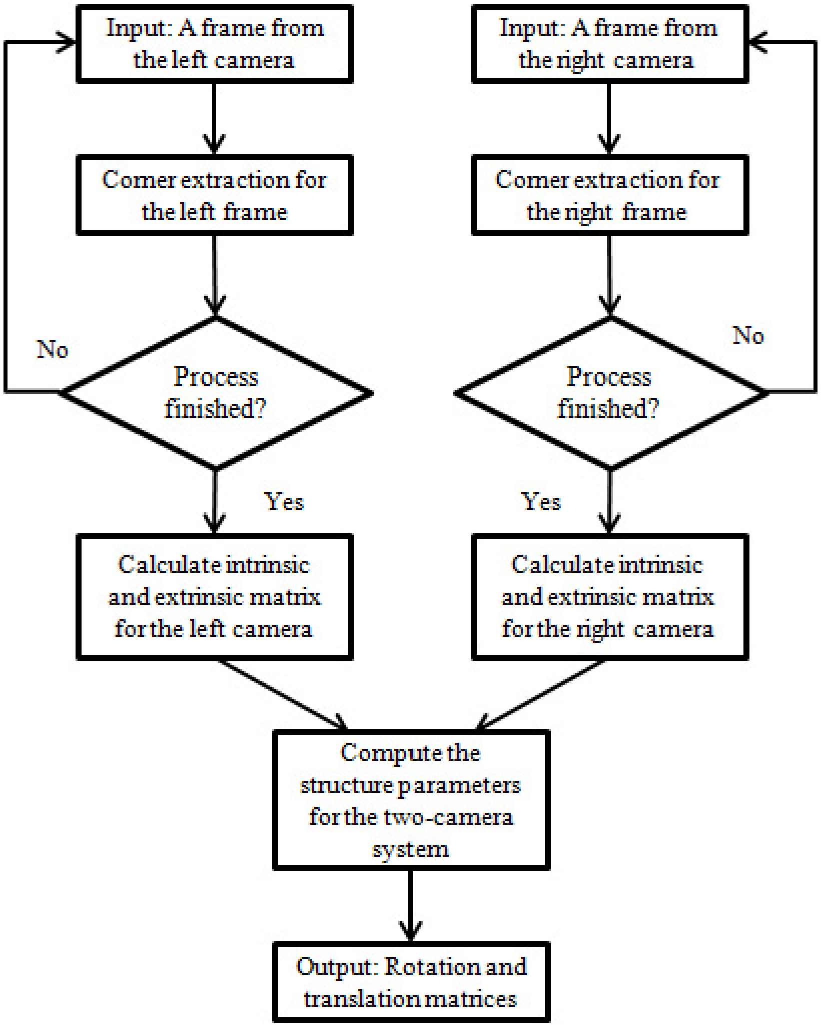

Input: A frame from the left camera
Input: A frame from the right camera
Corner extraction for the left frame
Corner extraction for the right frame
No
Process finished?
Process finished?
No
Yes
Yes
Calculate intrinsic and extrinsic matrix for the left camera
Calculate intrinsic and extrinsic matrix for the right camera
Compute the structure parameters for the two-camera system
Output: Rotation and translation matrices

Figure 6- 6 The procedure for two-camera calibration

In the proposed structure, the image processor model (Figure 6-6) plays an important role. The 3-D coordinates of the feature markers, attached to the catheter, were converted into a world coordinate system (WCS), using an image processor including camera rectification, feature extraction and stereo matching. The left and right calibrated cameras were rectified to improve the computing speed. Corresponding feature points from left and right images were extracted for stereo matching. The 3-D coordinates for the feature points were then calculated.

6.2.2 Two-camera System Calibration

Two-camera system calibration is the most basic step in stereoscopic vision. For stereo calibration, the two cameras were calibrated simultaneously; a rotation matrix and the associated translation vector were calculated to relate the two cameras.

For single-camera calibration, the camera intrinsic matrix and the distortion coefficients were computed as intrinsic parameters for the camera. Because the center of the chip is not, in general, positioned on the optical axis, two new parameters, cx and cy, were required to model the displacement coordinates with respect to the optic axis and center of the projection screen. In practice, we used three radial distortion terms, k1, k2, and k3. To characterize tangential distortion, we added two additional parameters, p1 and p2, as Eq. (2) shows:

$$\mathrm{x}' = \mathrm{x}(1 + \mathrm{k}_1\mathrm{r}^2 + \mathrm{k}_2\mathrm{r}^4 + \mathrm{k}_3\mathrm{r}^6) + 2\mathrm{p}_1\mathrm{xy} + \mathrm{p}_2(\mathrm{r}^2 + 2\mathrm{x}^2)$$

$$\mathrm{y}' = \mathrm{y}(1 + \mathrm{k}_1\mathrm{r}^2 + \mathrm{k}_2\mathrm{r}^4 + \mathrm{k}_3\mathrm{r}^6) + 2\mathrm{p}_2\mathrm{xy} + \mathrm{p}_1(\mathrm{r}^2 + 2\mathrm{y}^2)\text{(Eq. 6-2)}$$

(x, y) is the original coordinate of the distorted point on the imager, and *(x', y')* is the new location as a result of the correction.

To determine the intrinsic parameters of the cameras, each of the cameras was calibrated using Zhang’s method [Zhang02]. This method is simple and based mainly on different nonparallel views of a planar pattern, such as a chessboard. The calibration method used to compute intrinsic parameters for each of the cameras is described below.

1) First, two frames of a 6 × 8 black and white flat chessboard pattern were simultaneously captured from the left and right cameras (Figure 6-7).

2) The corners of the black and white squares in each of the frames were detected and used to calculate the exact locations of the corners to sub-pixel accuracy.

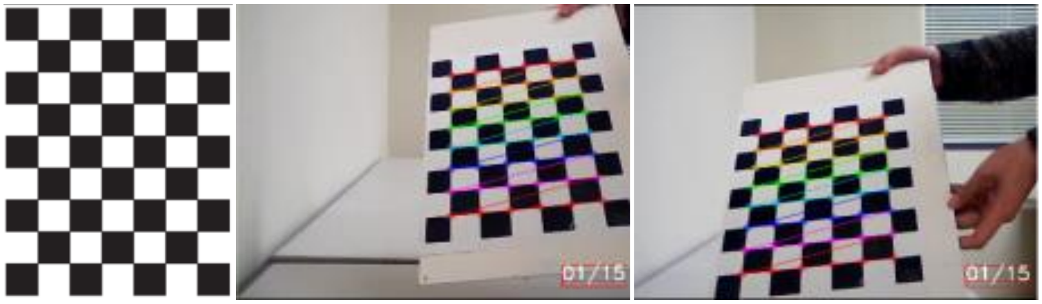
3) A homograph matrix was identified, based on two parts: (i) the physical transformation, which is the total effect of the rotation and translation that relates the chessboard pattern to the image plane; and (ii) the projection, which introduces the camera intrinsic matrix for all points in the set of images, based on a maximum-likelihood algorithm.

4) The cameras' intrinsic parameters were separated from the perspective projection matrix under the assumption of no camera distortion.

5) The extrinsic parameters for each image of the chessboard pattern were then determined, to describe the position of the chessboard pattern relative to the camera in each image. In practice, the extrinsic parameters are commonly specified by a rotation matrix and a translation vector. The extrinsic matrix translates the coordinates of a point in the camera's coordinate system to its corresponding point in the WCS.

6) After the intrinsic and extrinsic matrices were estimated, the

corresponding equations were solved to determine the distortion parameters



(a) A 6 ×8 chessboard pattern. (b) Two-camera calibration process.

Figure 6- 7 The chessboard used for two-camera calibration.

After each of the cameras was calibrated, a mapping relationship was constructed to connect the points in the physical world with the image points. Therefore, given a location in the 3-D physical coordinate frame attached to the camera, the corresponding point on the imager in pixel coordinates can be calculated, as specified in Eq. 6-2.

The camera matrix can also be used to perform the reverse operation; however, for a single-camera system, a point given in the image corresponds to a line in the physical 3-D world. Therefore, the specific correspondence relationship between points captured by one imager and those obtained by a second imager must be determined. Based on the correspondences and the baseline between the two cameras, the 3-D coordinates of the points in the physical world can be confirmed. Stereo calibration is essential to the computation of this geometrical relationship, and depends on the rotation matrix R and the translation vector T between the two cameras, as given by Eq. 6-3. Given several sets of joint views of chessboard corners, the rotation and translation parameters between the two cameras were computed. Additionally, according to the intrinsic matrices of both cameras, the fundamental matrix F, which relates the points on the image of one camera in pixel coordinates to the points on the image of the other camera in pixel coordinates, was calculated. After stereo calibration, the left camera was in the same plane with respect to the other camera.

$$\mathrm{P}_{\mathrm{right}} = \mathrm{R}(\mathrm{P}_{\mathrm{left}} - \mathrm{T}) \qquad \text{(Eq. 6-3)}$$

P_{right} *and* P_{left} were the observed locations of a point P on the two images.

6.2.3 Two-camera System Rectification

Assuming the fundamental matrix is known, then given a feature in one image, the corresponding epipolar (along which the matching point in the other image must lie) can be calculated; this is known as the epipolar constraint. The epipolar constraint makes it possible to convert a two-dimensional (2-D) search for matching features between two images into a one-dimensional (1-D) search along the corresponding epipolar line. This approach is computationally efficient, in that it can also refuse points that could result in erroneous correspondences.

The results indicated that it was better to reproject the image planes of the two cameras, such that they resided in the same plane, with image rows perfectly aligned into a frontal parallel configuration after rectification. By doing so, the stereo correspondence was more reliable and computationally tractable.

In the stereo rectification process, the results from stereo calibration, combined with the rotation matrix R and translation vector T, were used to rectify the two images. The epipolar lines were arranged along image rows. The scan lines were identical across both images; it was easier to generate depth information when the two image planes aligned exactly.

The two-camera system rectification method used in our approach is based on Bouguet's algorithm, which attempts to minimize the amount of image reprojection distortion for each of the two images, while maximizing the common viewing area. To minimize image reprojection distortion, the rotation matrix R, which represents the rotation relationship between the right camera’s image plane and the left camera’s image plane, was divided in half between the two cameras. In other words, each camera was rotated by half the rotation matrix R, such that their principal rays each ended up parallel to the vector sum of where their original principal rays had been pointing. Obviously, this kind of half rotation can put the two cameras into coplanar alignment; however, they are not row-aligned. Thus, the next step was to calculate a rotation matrix Rrect, which takes the two cameras’ epipoles to infinity and aligns the epipolar lines of the two cameras horizontally. R_{rect} consists of three parts: the vector corresponding to the direction of epipole e_l, the second epipole vector, e_2, and the vector orthogonal to both; i.e., the cross product of e_1 and e_2. Assuming that the principal point (c_x, c_y) corresponds to the center point of the left image, the unit normalized epipole vector is directed along the translation vector between the two cameras' centers of projection: $e_1 = T/||T||$. The second vector, e_2, must be orthogonal to e_1. This can be accomplished using the cross product of e_1 with respect to the direction of the principal ray. The cross product can then be normalized as the second unit vector: $e_2 = \left[-T_y\ T_x\ 0\right] / \sqrt{{T_x}^2 + {T_y}^2}$. The third vector e_3 is orthogonal with respect to both unit vectors, e_1 and e_2; i.e., the cross product, $e_3 = e_1 \times e_2$.

The rotation matrix R_{rect}, which can take the epipole in the left camera to infinity, is then given by $R_{rect} = [(e_1)^T\ (e_2)^T\ (e_3)^T]^T$. This matrix rotates the left camera about the center of the projection, such that the epipolar lines become horizontal and the epipoles are at infinity. Row alignment of the two cameras was then achieved using the following setting:

$$\mathrm{R_{left}} = \mathrm{R_{infinity}} \mathrm{r_{left}}$$
$$\mathrm{R_{right}} = \mathrm{R_{infinity}} \mathrm{r_{right}} \qquad \text{(Eq. 6-4)}$$

where r_{left} and r_{right} represent the half rotation matrices for the left and right cameras, respectively, and R_{left} and R_{right} represent the rectification transforms for the left and right cameras, respectively.

The projection matrix in the rectified coordinate system for the left and right cameras, P_{left} and P_{right}, respectively, can be determined using the following:

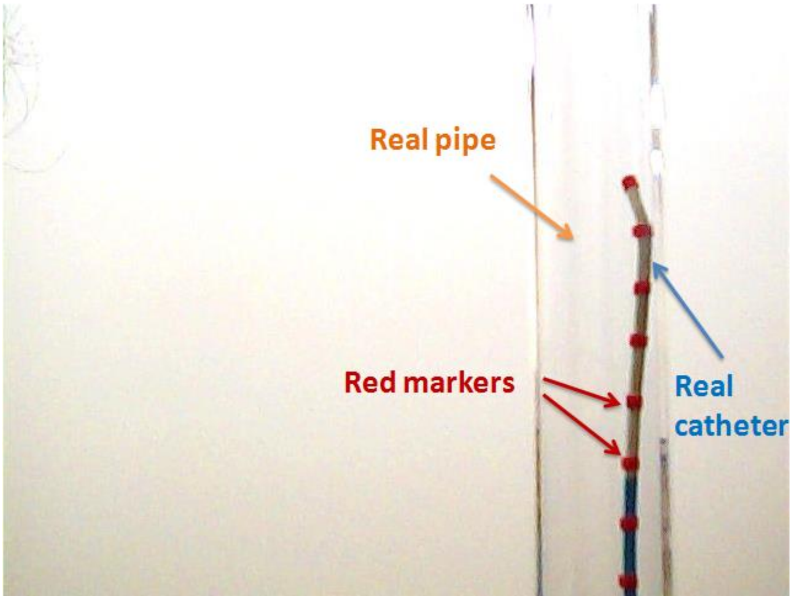
$$P_{left} = \begin{bmatrix} f & 0 & c_{rectified\_x} \\ 0 & f & c_{rectified\_y} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

$$P_{right} = \begin{bmatrix} f & 0 & c_{rectified\_x} \\ 0 & f & c_{rectified\_y} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & T_x \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad \text{(Eq. 6-5)}$$

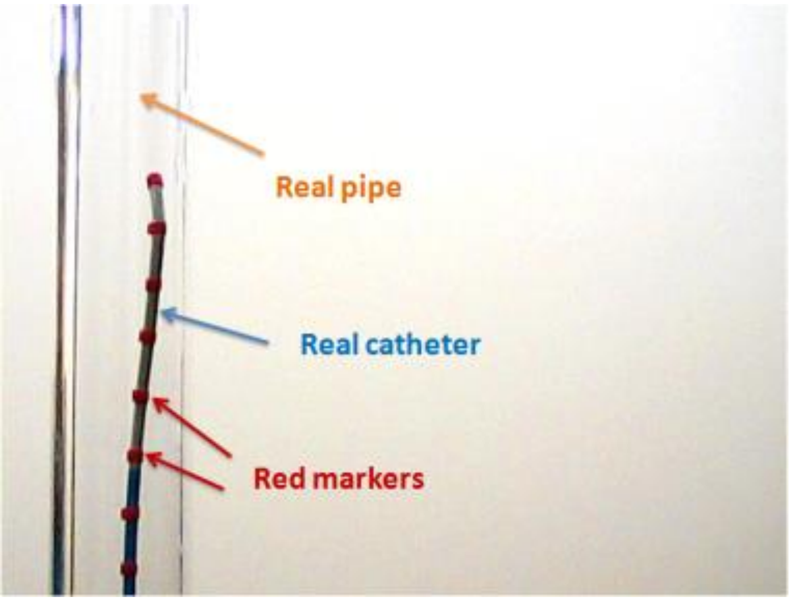
where T_x is the horizontal shift between the cameras and ($c_{rectified\_x}$, $c_{rectified\_y}$) is the principal point on the two rectified images.

The left and the right camera views were shifted relative to each other mainly along the x-axis. In the rectified images, the corresponding epipolar lines in the left and right cameras were horizontal and had the same y-coordinate.

Figures 6-8 (a) and (b) show the original pictures captured simultaneously by the left and right cameras, respectively; a pair of rectified images, resulting from the proposed algorithm from each of the cameras is shown in Figure 6-9. The green lines in Figure 6-9 represent the row-aligned situation in the two rectified images.

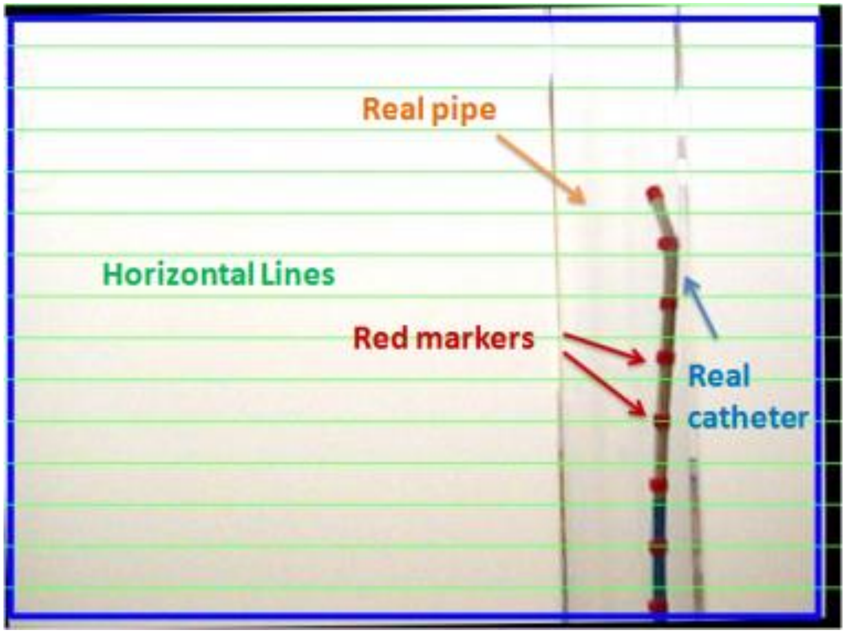


(a) Left camera image.

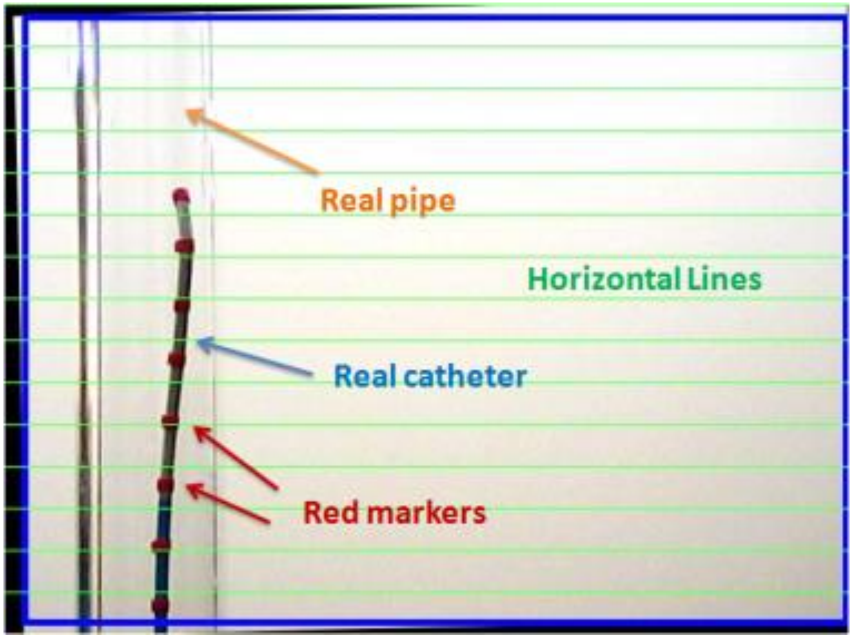


(b) Right camera image.

Figure 6- 8 The original pictures captured by the two-camera system



(a) Rectified image from left camera.



(b) Rectified image from right camera.

Figure 6- 9 The rectified images based on the proposed algorithm.

6.2.4 Feature Extraction and Stereo Matching

To shorten the stereo correspondence time and make it easier to compute the disparity between two features, red instrument markers were attached to a 2-mm-diameter catheter; every two red markers deployed had a 1-cm-spacing. The specific marker distribution is shown in Figure 6-12. The length of a real catheter is ~50 cm; thus, we used 50 red markers to represent the catheter shape. The shape of the catheter inside the pipe was verified by referring to the calculation of the 3-D coordinates for the 50 red markers. Note that the markers were extremely thin, and therefore, did not enlarge the diameter of the real catheter. Additionally, lubricants were used with the catheters in both the real and virtual surgical procedures to decrease frictional resistance.

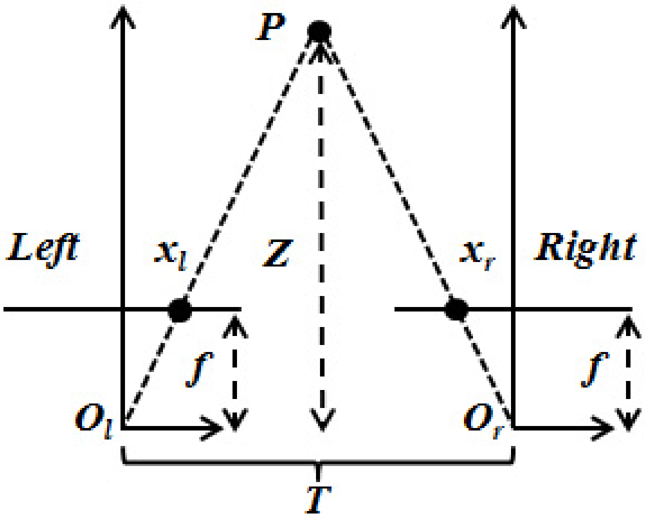


Figure 6- 10 The stereo figure for the two cameras after stereo rectification

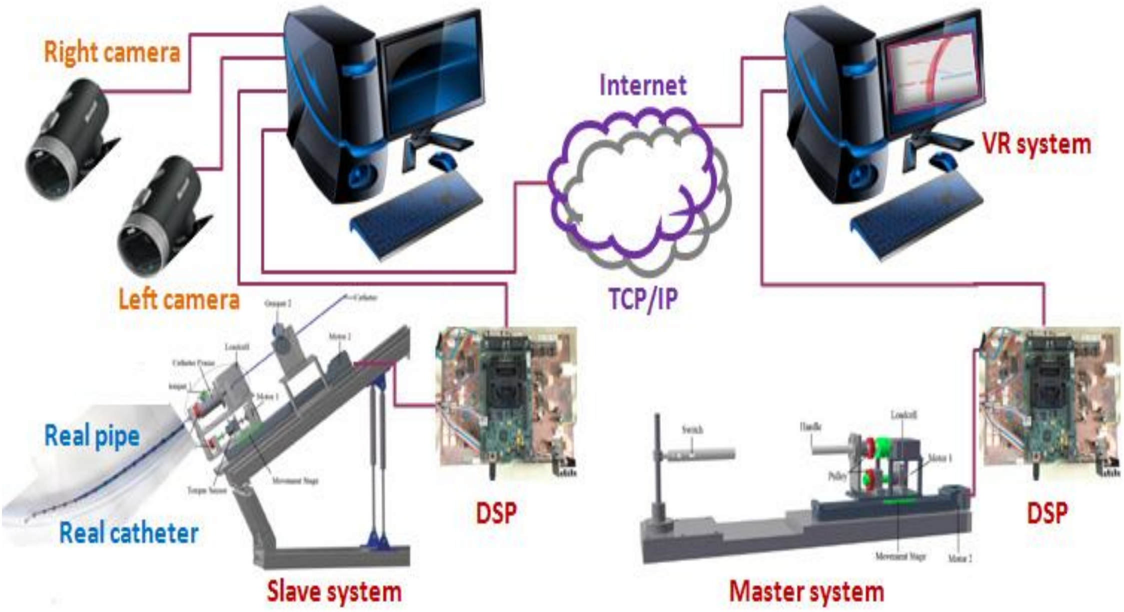


Figure 6- 11 The surgical procedure using the proposed catheter operating system

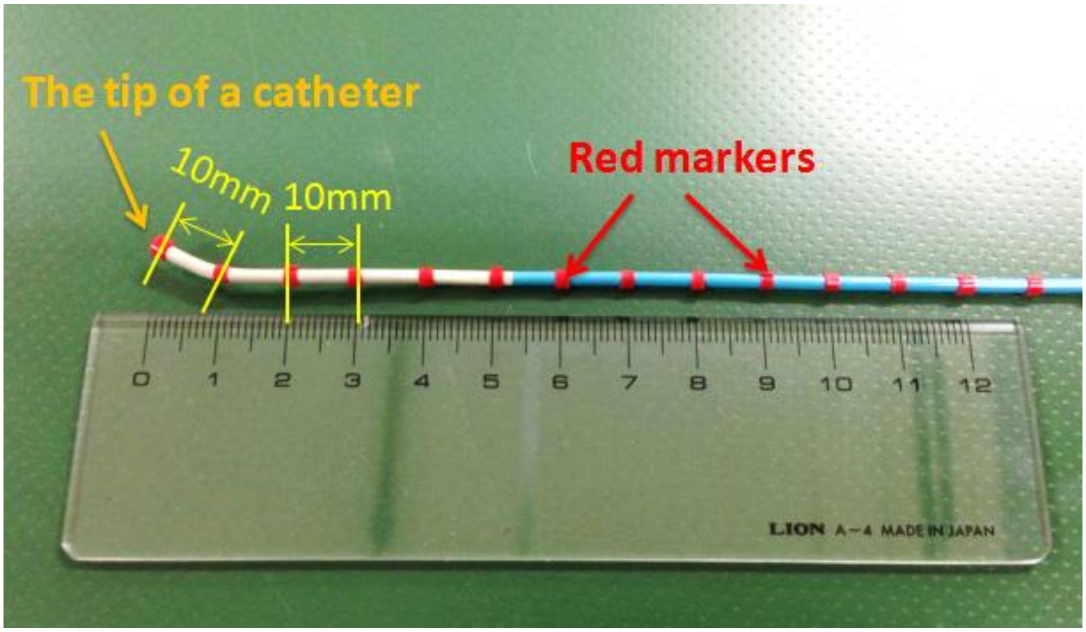
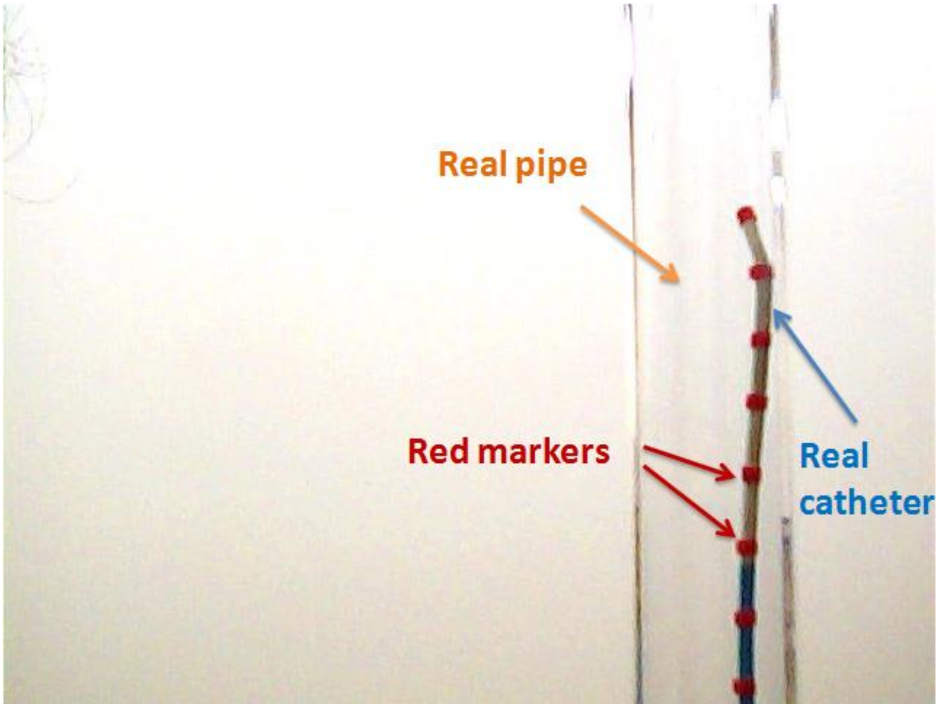
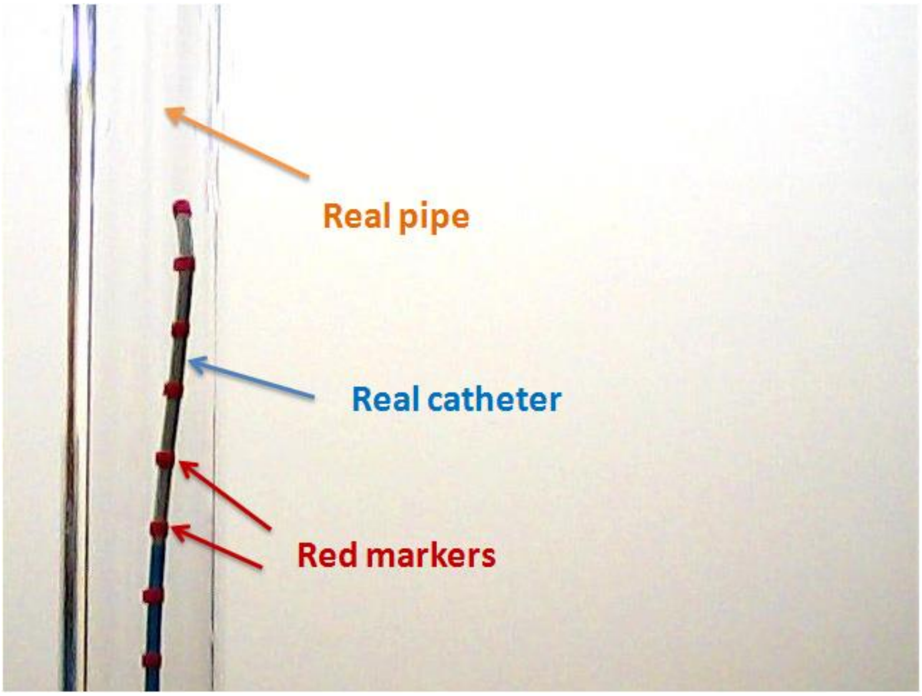


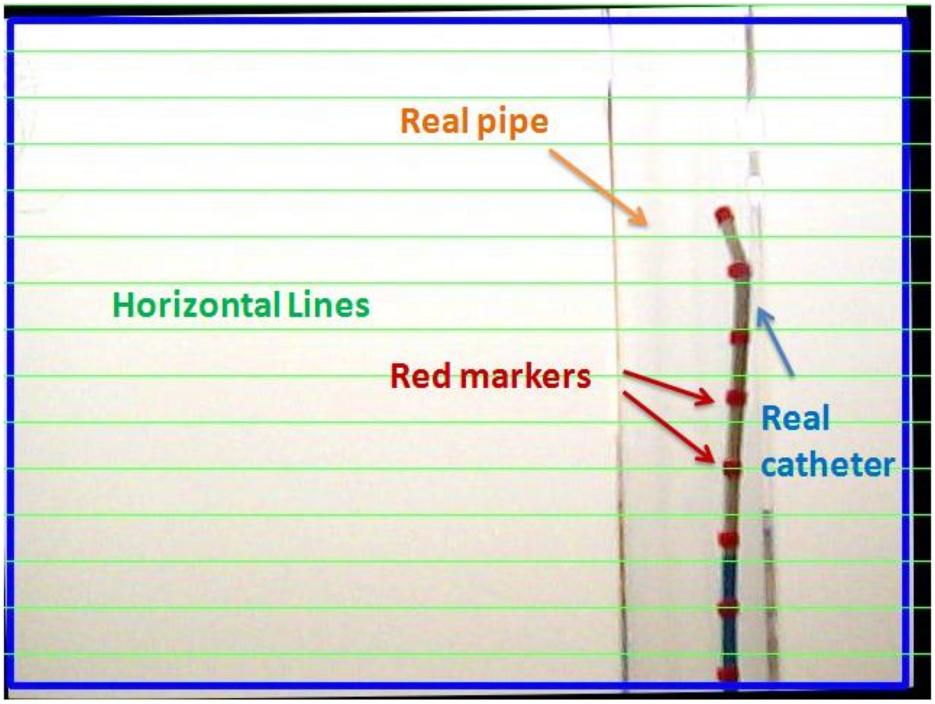
Figure 6- 12 The red markers attached to a 2-mm-diameter catheter



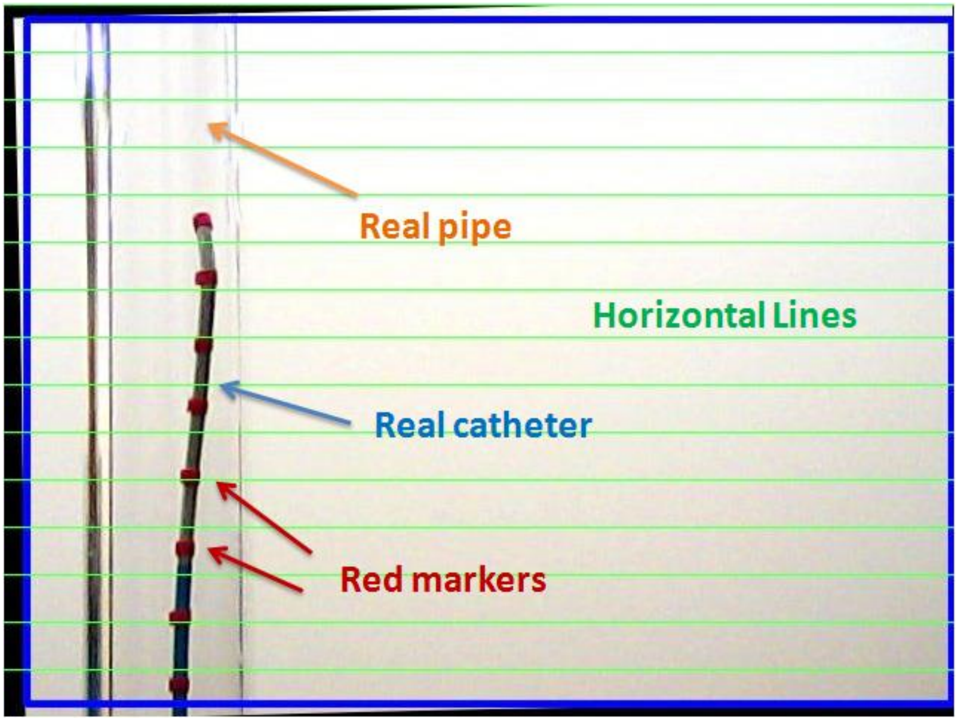
(a) Original image from left camera



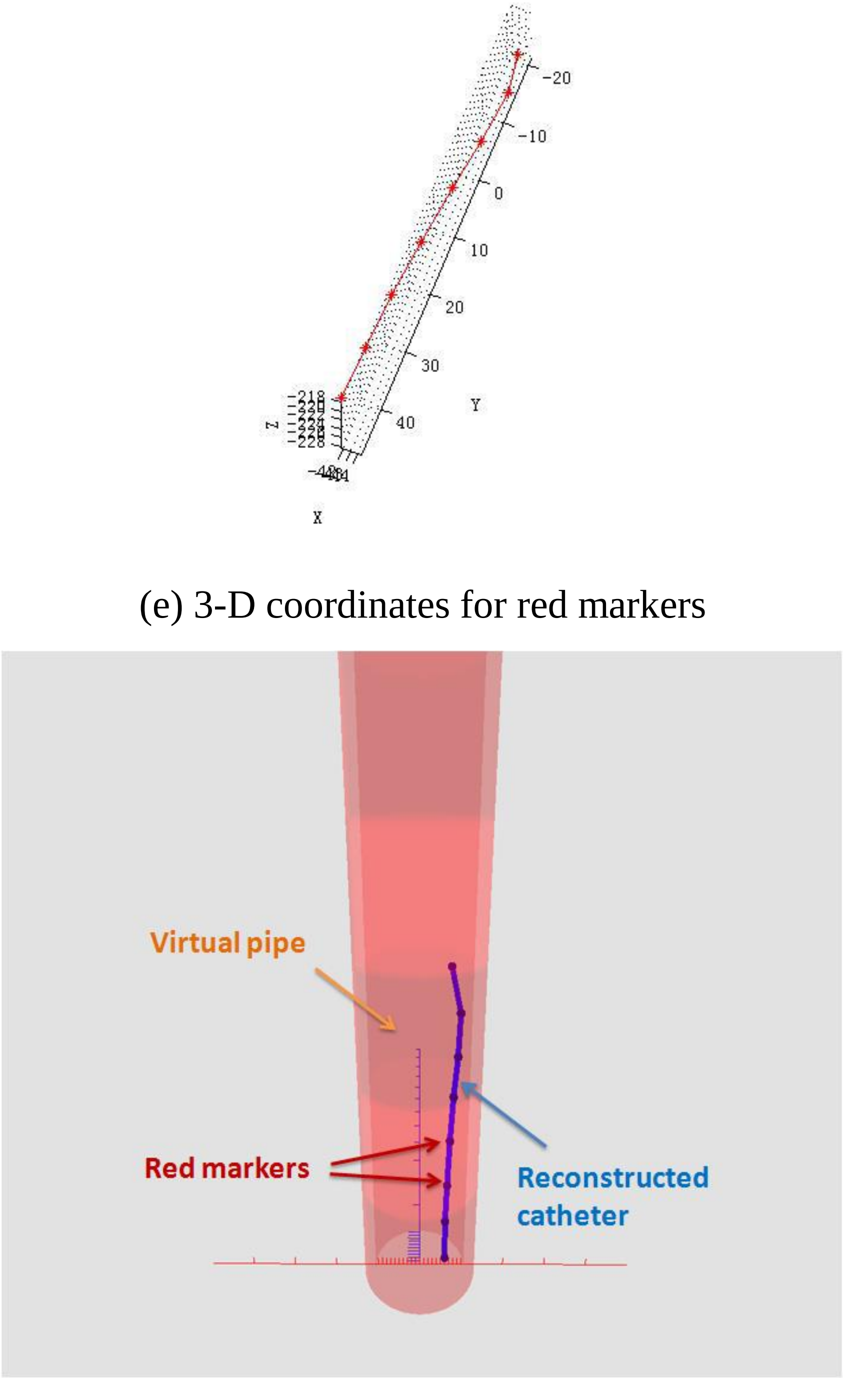
(b) Original image from right camera



(c) Rectified image from left camera



(d) Rectified image from right camera



(e) 3-D coordinates for red markers

(f) Reconstructed catheter in virtual pipe

Figure 6- 13 The rectified images based on the proposed algorithm

To recover the 3-D coordinates of each marker on the catheter, we first needed to extract the corresponding feature points (red markers). The feature points from the two cameras were then matched to form identical points, and the disparity between two corresponding feature points was calculated. To identify the red markers in two rectified images, simultaneously captured from the left and right cameras, the images were first converted from red-green-blue (RGB) format to hue-saturation-value (HSV) format. The red markers were then separated from the images by a chain of image processing algorithm; then, lower and upper threshold values were used to extract red markers from the background. Next, an erosion and dilation algorithm was utilized to remove the noise generated by the experimental environment. Finally, circumcircles were calculated for every red marker; the coordinates of the center of each circumcircle were used as its 2-D pixel coordinates.

Because the image rows of the two cameras were aligned after rectification, matching points for the left and right cameras were located in identical horizontal positions and had identical vertical coordinate values. Additionally, if points Pa and Pb appeared vertically in a certain order in one image, then these two points occurred vertically in the same order in the other image. The red markers were visible in both the left and right images, and were the same in number. This facilitated feature point (red marker) matching in both images along the vertical coordinate; i.e., a red marker in the left image must have a corresponding marker that lies practically along the same horizontal line in the right image. For best computation, the red marker's y-coordinate in the right-camera image was adjusted to be the same as that in the left-camera image.

6.2.5 3-D Coordinate Reconstruction

The processes of stereo calibration and stereo rectification of the two cameras resulted in a pair of undistorted, row-aligned, measured cameras, whose image planes were coplanar with respect to the other. Additionally, every pixel row of the left camera was aligned with the corresponding row in the right camera. The origins of the left and right images are defined at the upper left part. Ol and Or represent the centers of the projection, and the principal rays intersect the image planes at the principal points. The cameras, having the same focal length f were then displaced from one another by T. Based on the rectified model for the left and right cameras (Figure 6-10), the depth information Z can be easily derived, using similar triangles:

$$Z = \frac{fT}{x_l - x_r} \qquad \text{(Eq. 6-6)}$$

Most importantly, the reprojection matrix Q, which can project a 2-D point in the pixel coordinate system to a 3-D point in WSC, can be computed as follows, in accordance with the calibration and rectification results:

$$Q = \begin{bmatrix} 1 & 0 & 0 & -c_x \\ 0 & 1 & 0 & -c_y \\ 0 & 0 & 0 & f \\ 0 & 0 & -1/T_x & 0 \end{bmatrix} \qquad \text{(Eq. 6-7)}$$

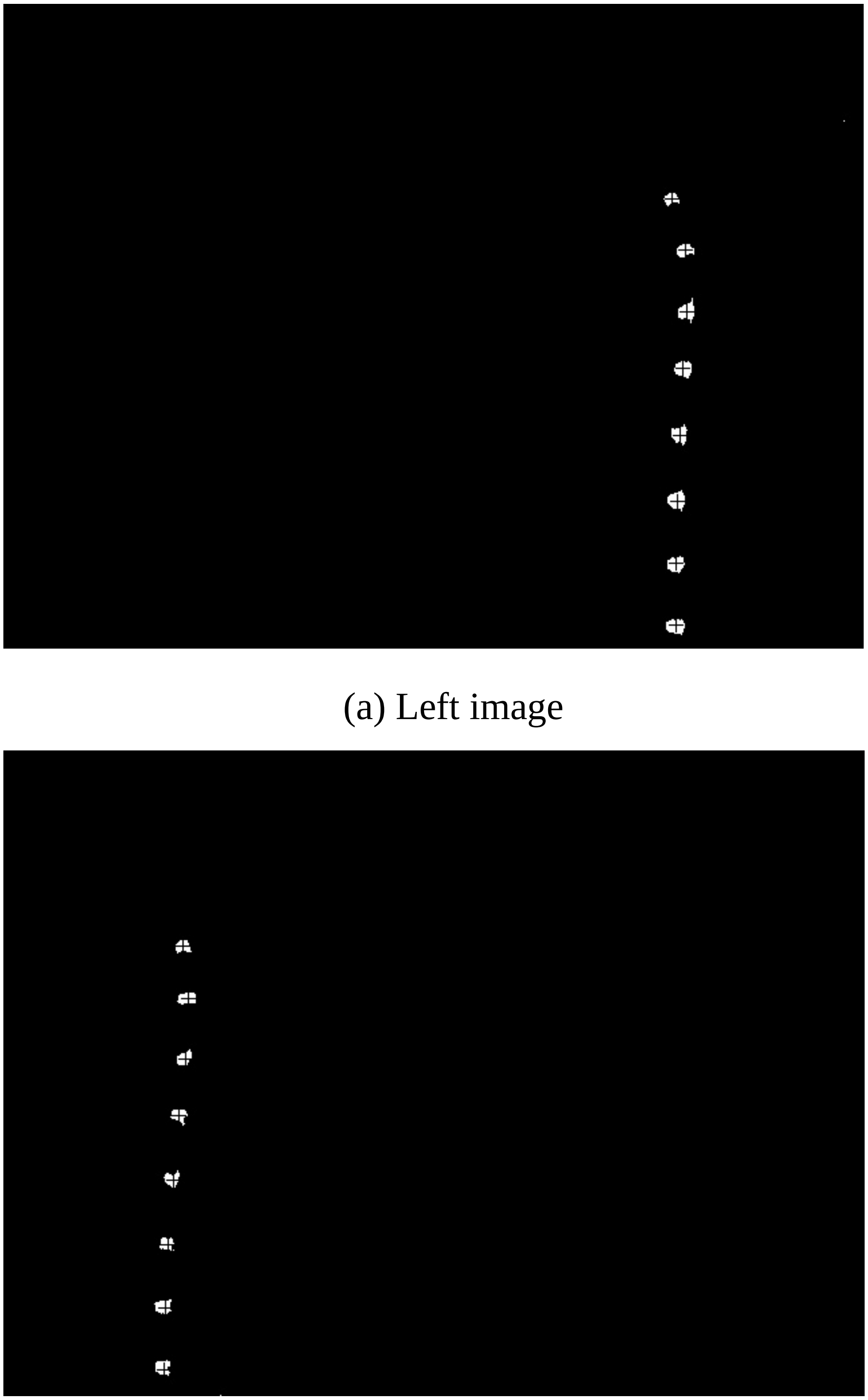
Note that the parameters were generated from the left image. The 3-D coordinates can then be determined from the reprojection matrix, a 2-D homogeneous point, and its associated disparity d:

$$Q \begin{bmatrix} x \\ y \\ d \\ 1 \end{bmatrix} = \begin{bmatrix} X \\ Y \\ Z \\ W \end{bmatrix} \qquad \text{(Eq. 6-8)}$$

where (X/W, Y/ W, Z/ W) represent the 3-D coordinates.

According to Eq. 6-6 and the pixel coordinates of the red markers in both the left and right images, the 3-D coordinates for the red markers can be converted into world coordinates.

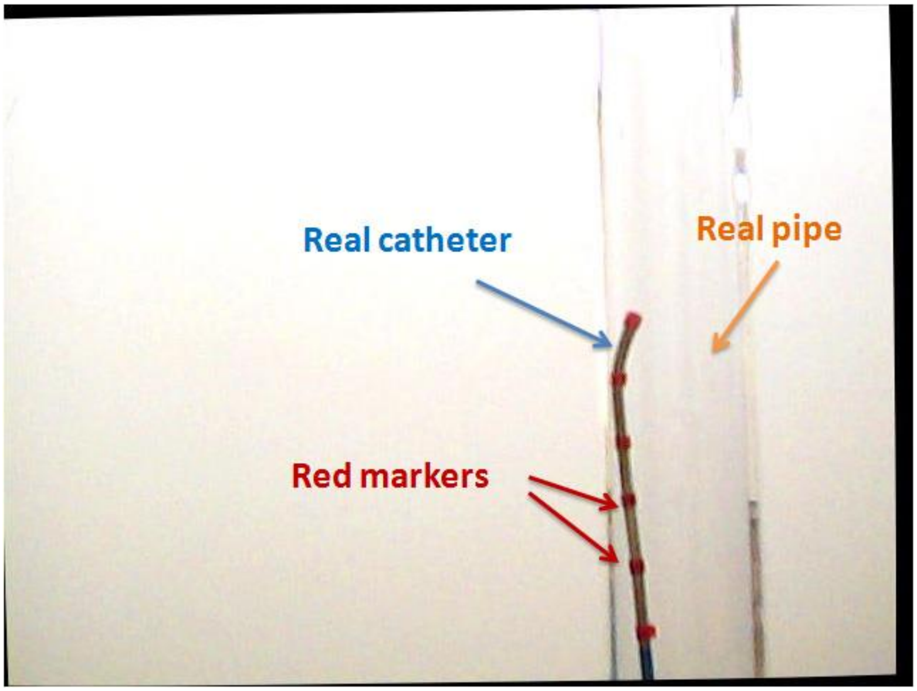
The 3-D reconstruction results for the red markers in Figure 6-13 (a) are listed in Figure 6-13 (e). The 3-D coordinates of the red markers were used to re-construct the catheter shape inside a virtual pipe at the operation location (as visual feedback). The virtual pipe had the same shape and size as the real pipe at the patient’s location, representing a blood vessel (Figure 6-13 (f)). The 3-D coordinates were transmitted from the patient’s location to the operation location via the Internet, as opposed to conventional visual information directly generated by a camera.



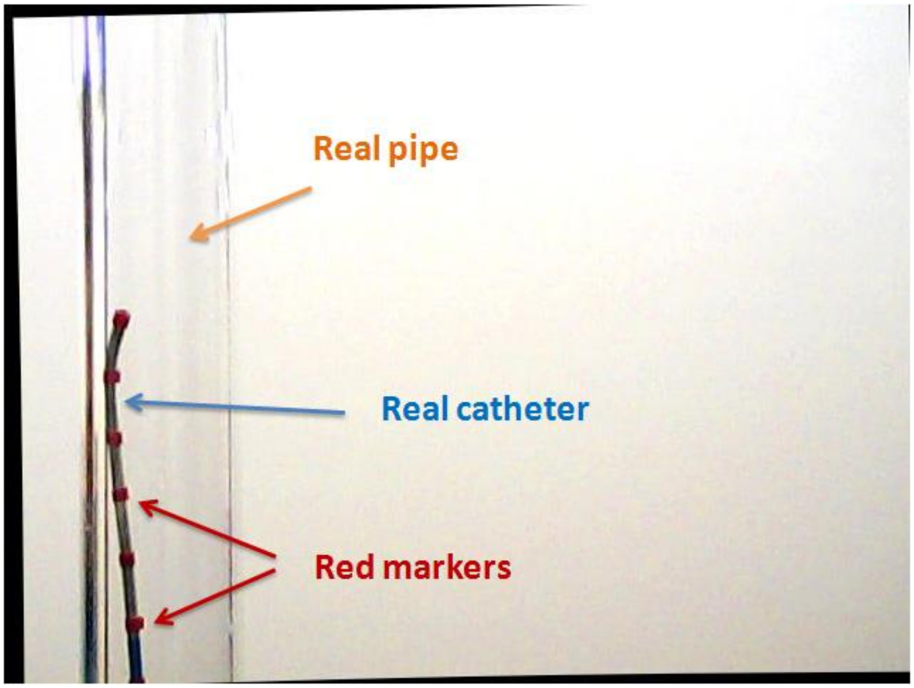
(a) Left image

(b) Right image

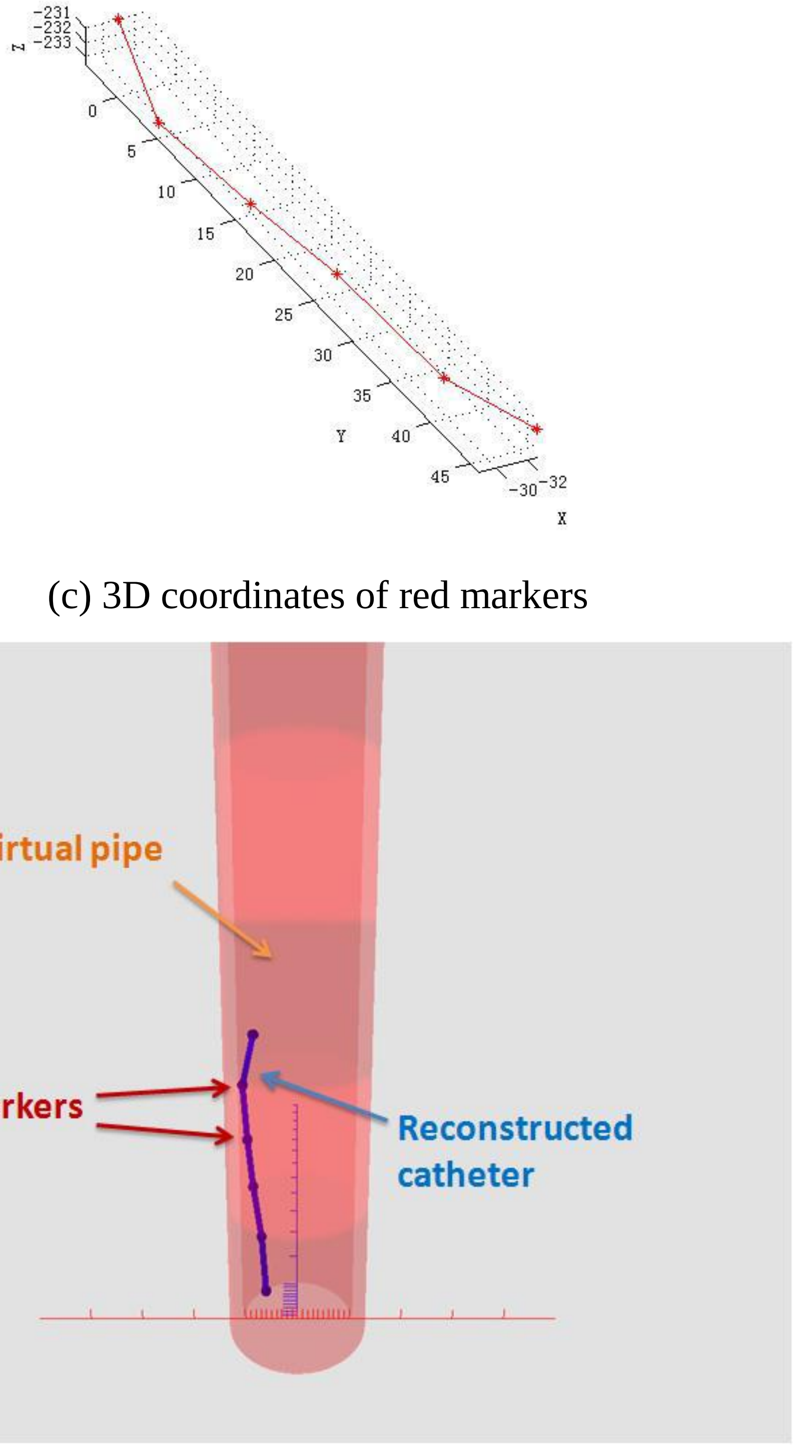
Figure 6- 14he extraction results for the red markers



(a) Rectified results from left frame



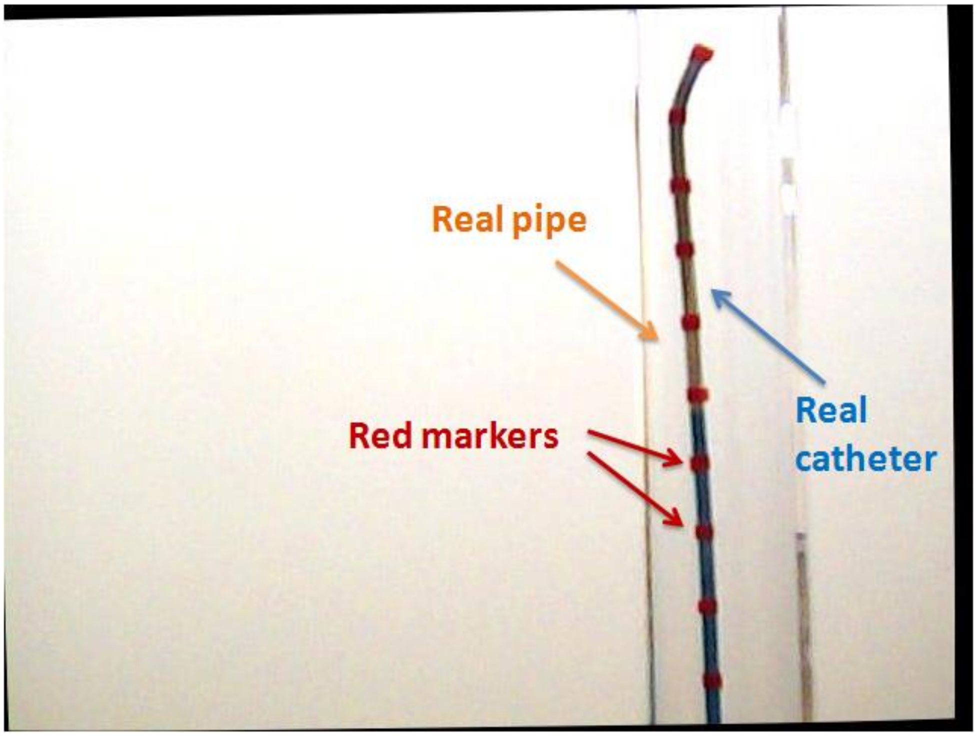
(b) Rectified results from right frame



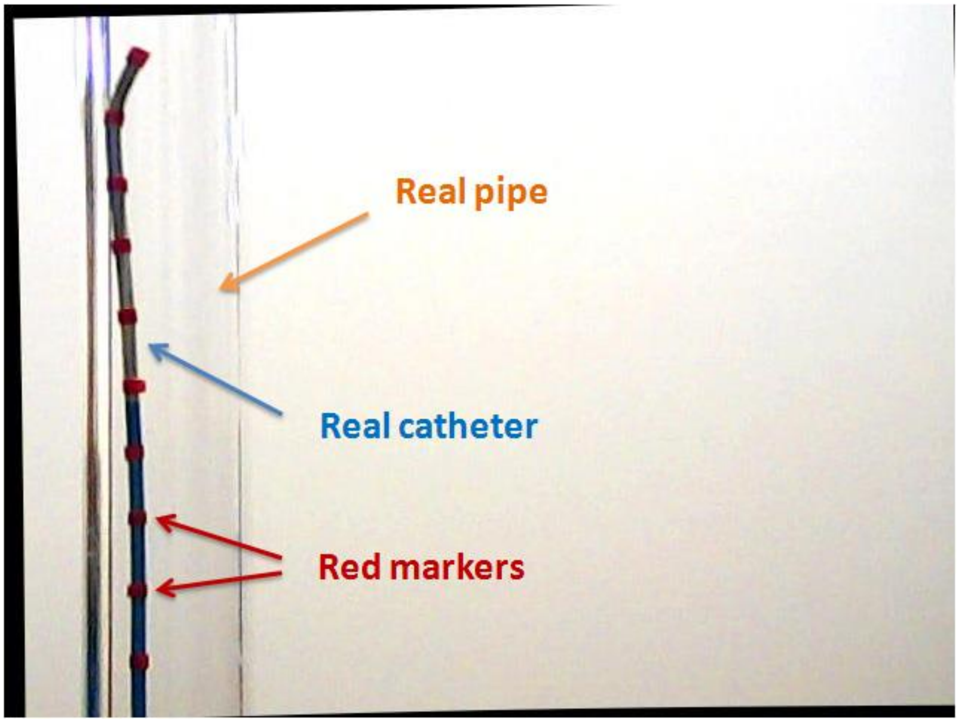
(c) 3D coordinates of red markers

(d) Reconstructed catheter in virtual pipe

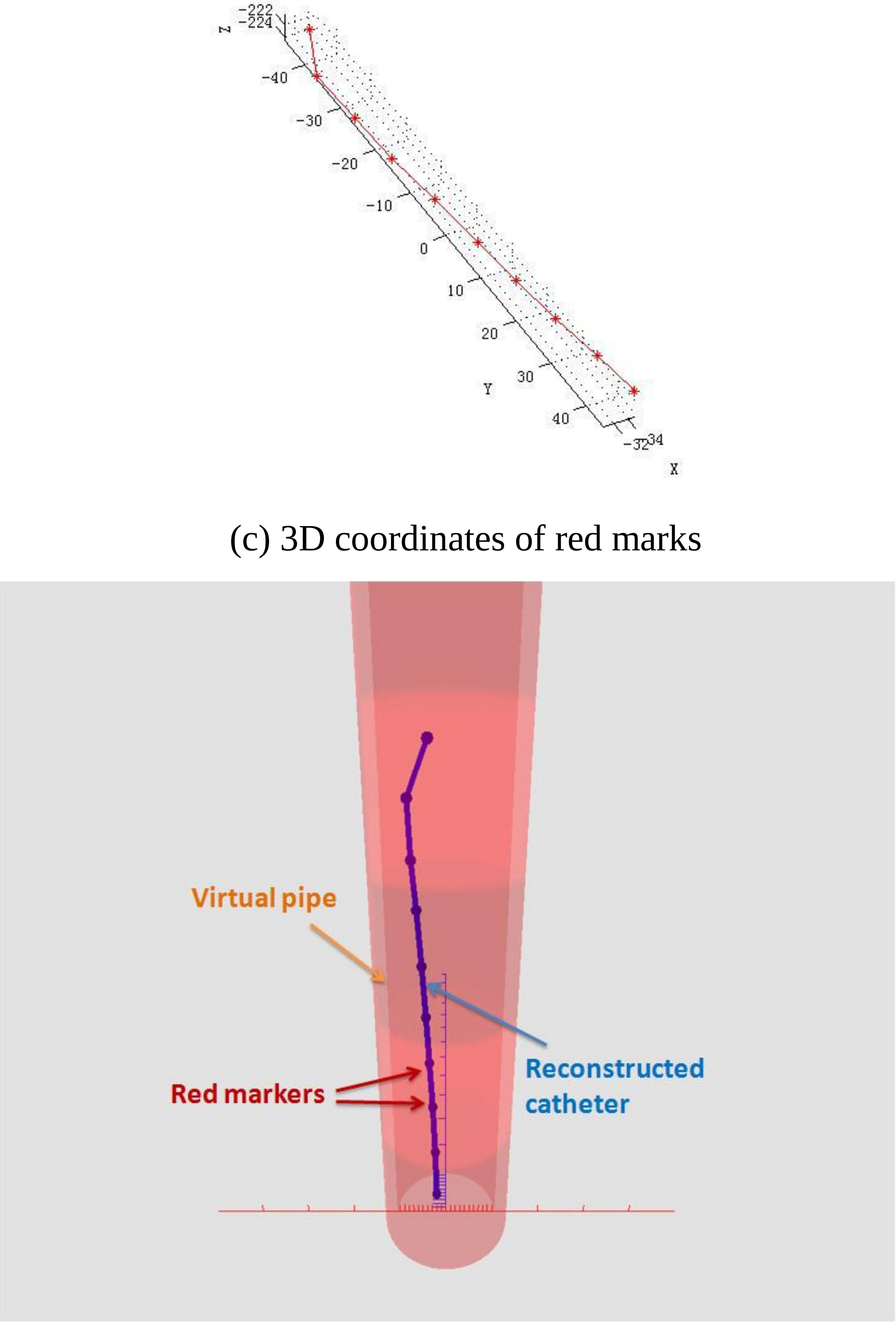
Figure 6- 15The insertion experiment inside a straight pipe-Frame 1



(a) Rectified results from left frame



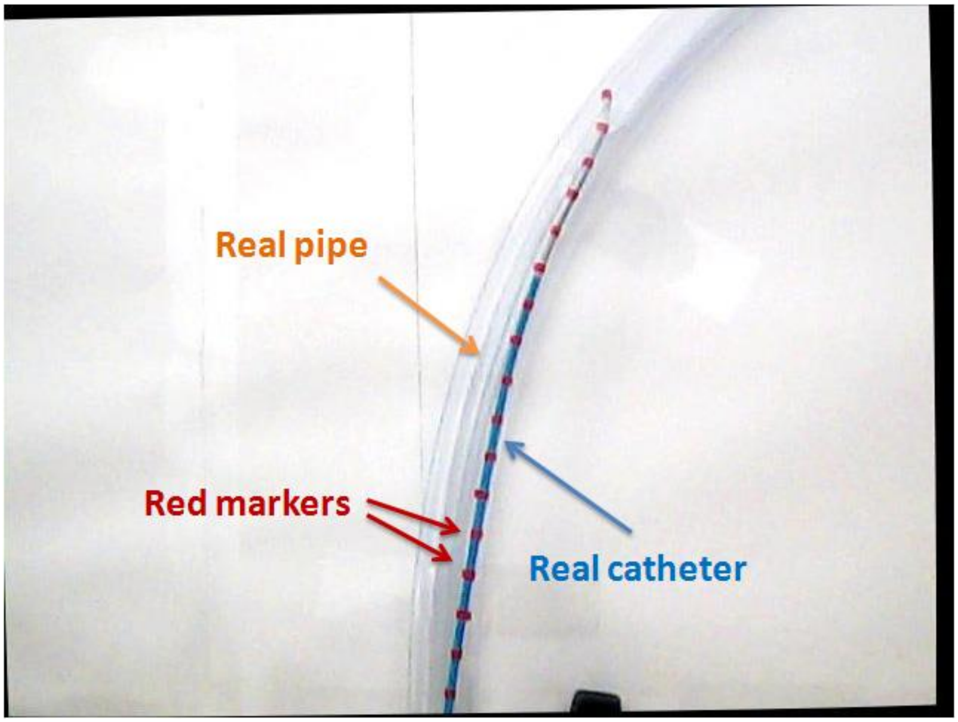
(b) Rectified results from right frame



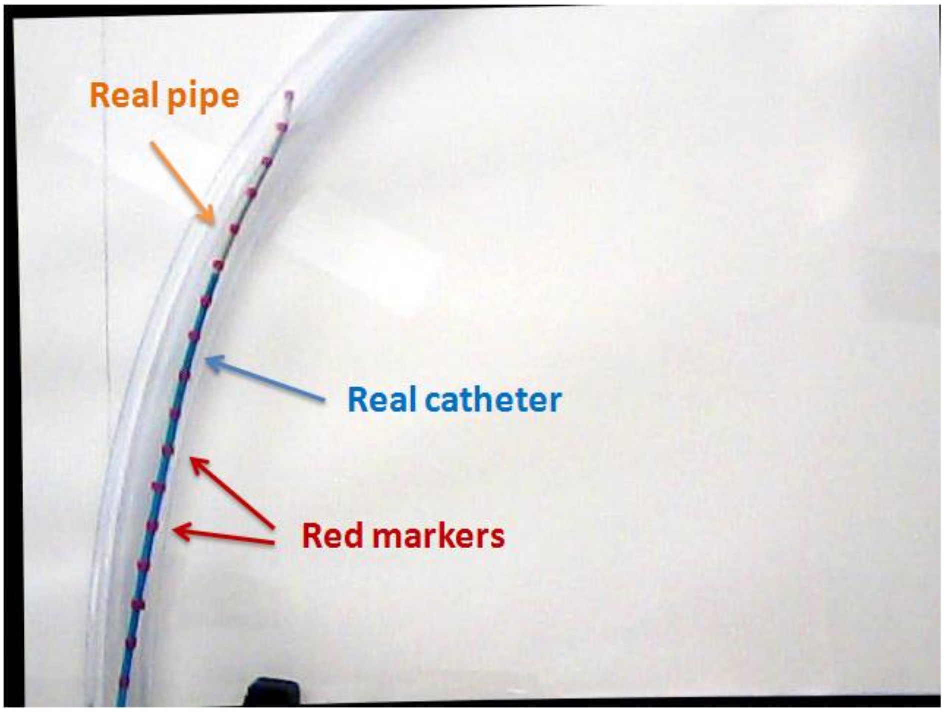
(c) 3D coordinates of red marks

(d) Reconstructed catheter in virtual pipe

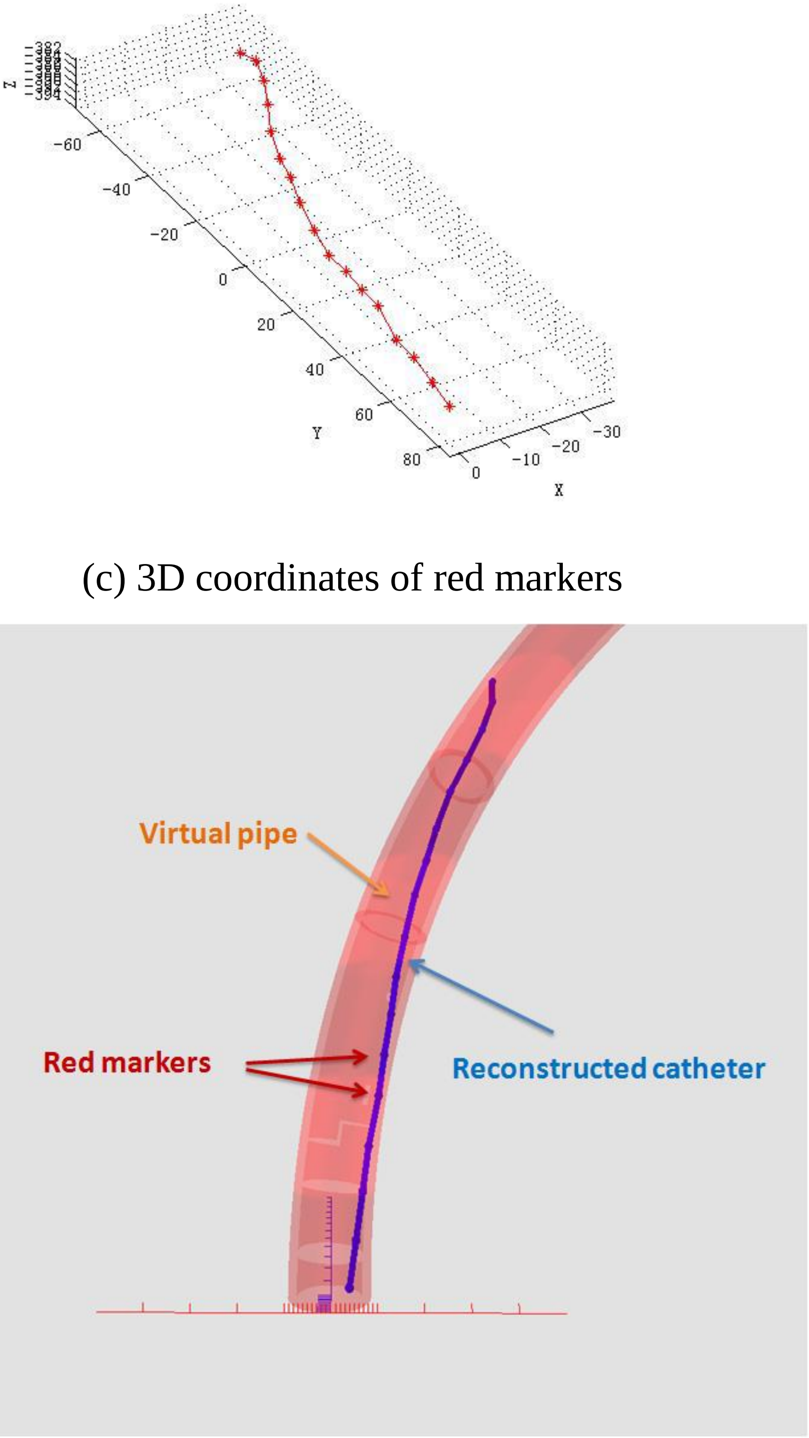
Figure 6- 16 The insertion experiment inside a straight pipe-Frame3



(a) Rectified results from left frame



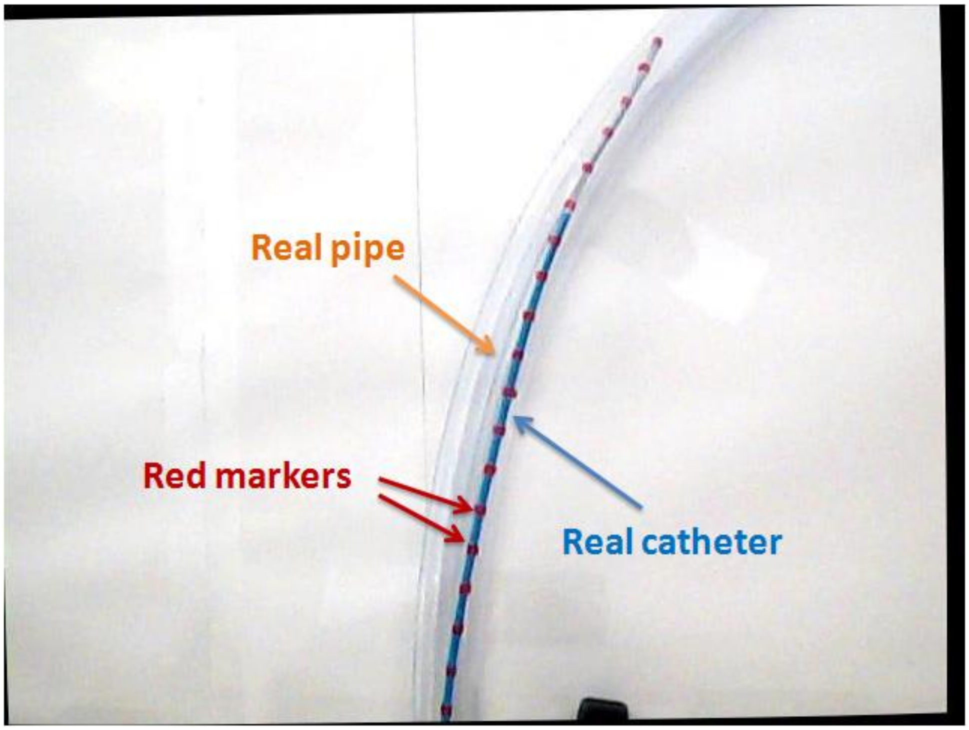
(b) Rectified results from right frame



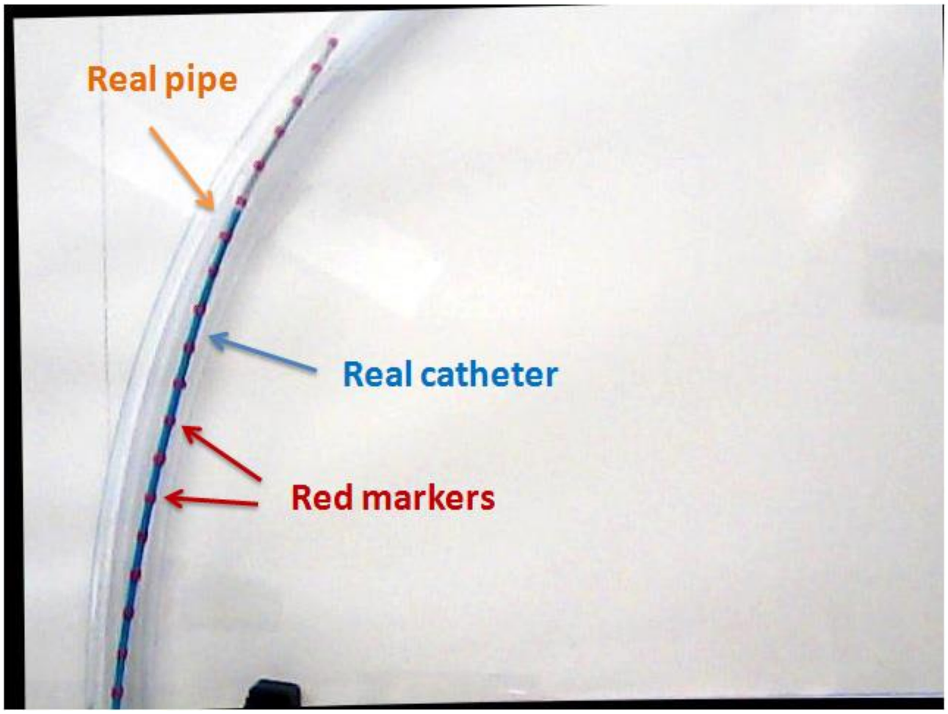
(c) 3D coordinates of red markers

(d) Reconstructed catheter in virtual pipe

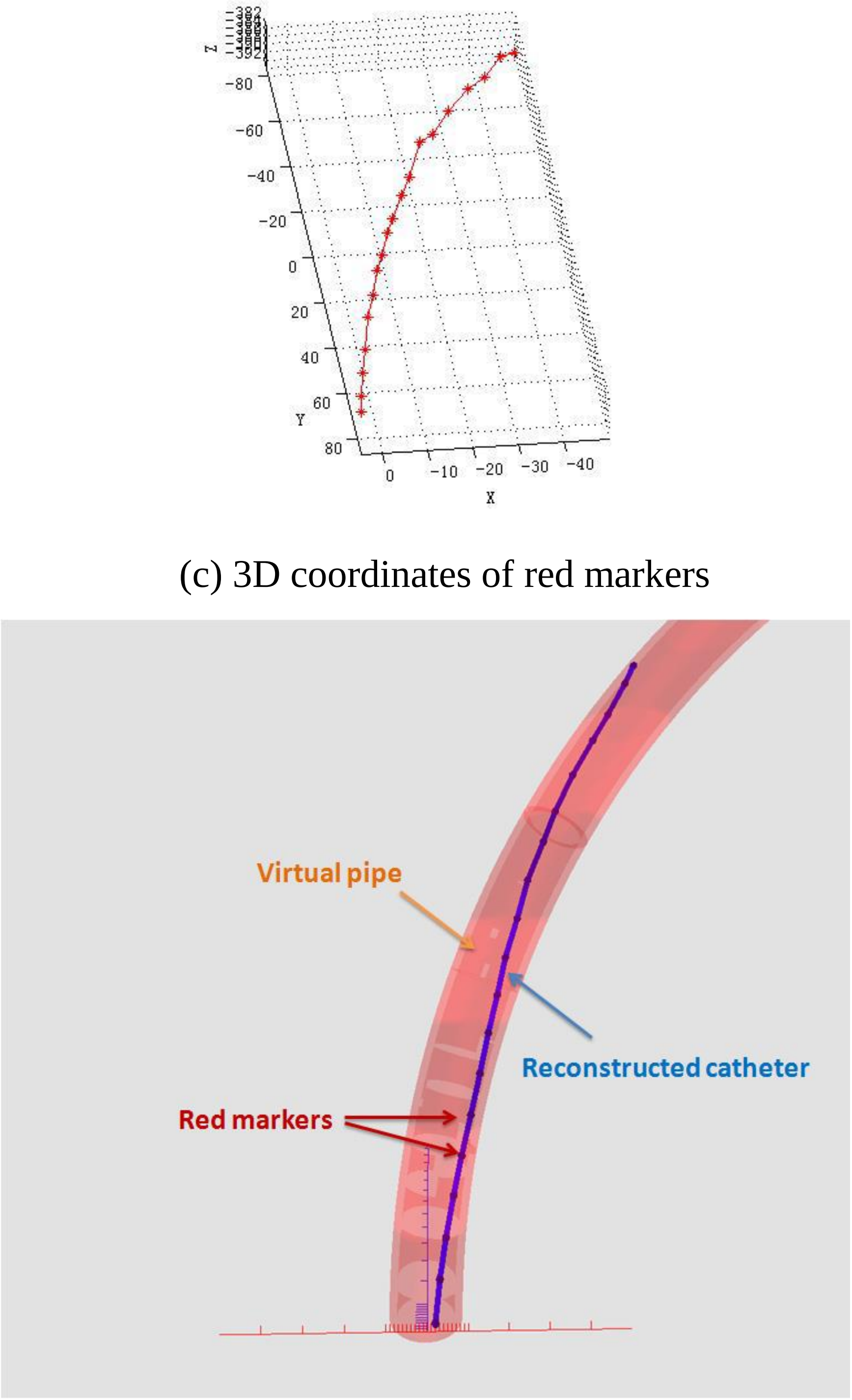
Figure 6- 17 The insertion experiment inside a bent pipe-Frame 1



(a) Rectified results from left frame



(b) Rectified results from right frame



(c) 3D coordinates of red markers

(d) Reconstructed catheter in virtual pipe

Figure 6- 18 The insertion experiment inside a bent pipe-Frame 2

6.2.6 Experimental setup

Internet-based tele-operative surgical systems have been used for remote experiments; the surgical system used for this study consisted of a master system and a slave system.

The catheter manipulator/insertion mechanism of the slave part of the entire system was located at the patient’s location. This system was equipped with two degrees of freedom (DOFs): axial and radial. Two graspers were used for axial and/or radial displacement. Grasper 1 was used by the surgeon to integrate the catheter with the manipulator. The manipulator can then drive a catheter along both the axial and radial direction. And the axial movement was realized by a movement stage. The movement stage was driven by a screw and stepping motor. Additionally, a direct current motor was coupled to the catheter frame by two pulleys, which were coupled by a belt with teeth. Motor 1 was used to move the catheter radially. Also, the haptic feedback in the axial direction was detected by the load cell when there was a collision between the catheter and the blood vessels. Additionally, the haptic feedback in the radial direction was detected by torque sensor.

When the catheter was clamped by Grasper 2, catheter retained its position, however the catheter drive was allowed to move freely.

The master system, as a surgeon’s controller, was located at the operation location. Surgeons operated the controller by pulling/pushing or rotating the right handle part. The axial and radial movements of the surgeon’s hand were measured by the right handle and were then transmitted to the catheter manipulator at the patient’s location. A switch fixed on the left handle part was used to control the grasper 1 and grasper 2 on the catheter manipulator. To allow the catheter manipulator to remain synchronized with the controller, the catheter insertion/rotation mechanism of the controller mimicked that of the catheter manipulator. The radial haptic feedback, detected by the loadcell fixed on the catheter manipulator, was displayed depending on the translational speed of the movement stage. Additionally, the direct current motor functioned in tandem with the control mode to generate a damping force to the surgeon according to the haptic feedback in the radial direction detected by torque sensor attached to the catheter frame at the slave system side.

To improve the accuracy of rotation angle and displacement between the catheter manipulator of the slave system and the controller of the master system, dynamic models and control methods (Fuzzy Proportional-Integral-Derivative) were used during remote surgical experiments.

Tele-operative animal experiments were carried out between Takamatsu, Japan and Beijing, China.

The problems with remote surgical procedures, to this point, were mostly related to the excessive transmission time required for visual feedback; this resulted in deterioration in operability and prolonged surgical times (i.e., significant radiation exposure). The purpose of our study was to develop a new method for remote surgical procedures involving catheter insertion/rotation, which would reduce the transmission time of visual feedback. As described above, our proposed method was based on an image processing algorithm and virtual reality technology. In the proposed approach, conventional image feedback was converted into 3-D coordinate feedback via markers positioned along the catheter. The experimental setup to test the 3-D reconstruction during remote surgery is shown in Figure 7-10. Two cameras (frame rate: 30 frames per second; dynamic resolution: 640 × 480) were connected to a PC at the patient's location and placed in the same plane, as much as possible, to capture the images of a catheter inside a plastic tube (representing a blood vessel). An image processing algorithm model was run on the PC at the patient's location to compute the 3-D coordinates of the red markers attached to the catheter during insertion/rotation of the catheter inside the tube by the slave system. These 3-D coordinates were then transmitted to the PC at the operation location via the Internet, and reconstructed within a corresponding 3-D simulated vessel model, having the same size and shape as the plastic tube in the virtual reality environment. Physicians controlled the master system at the operation location and carried out the insertion/rotation operation, based on the force feedback and visual feedback from the virtual reality environment.

6.2.7 Experimental results

We designed a series of experiments to test the accuracy of the reconstruction algorithm and the transmission time for visual feedback from the master system to the slave system, before and after applying the proposed architecture.

The distortion parameters, k_1, k_2, p_1, p_2, k_3, and the intrinsic parameters, f_x, f_y, c_x, c_y, α, resulting from Zhang's algorithm for each of the cameras, are listed in Tables I and II, respectively. The accuracy of this calibration method is reportedly 0.4836 pixels.

The distortion parameters and the intrinsic parameters listed in table I and table II were used to finish the two-camera rectification. Figure 6-14 (a) and (b) show the original pictures captured simultaneously by the left and right cameras, respectively; a pair of rectified images, resulting from the proposed algorithm from each of the cameras is shown in Figure 6-14 (c) and (d). The green lines in Figure 6-14 (c) and (d) represent the row-aligned situation in the two rectified images.

Two rectified images were processed by a series of proposed image processing algorithms to separate the corresponding feature points from the images. Figure 6-14 shows the extraction results for the red markers displayed in Figure 6-15 (c) and (d). According to Eq. 6-6 and the pixel coordinates of the red markers in both the left and right images, the 3-D coordinates for the red markers can be calculated and reconstructed in the virtual reality environment, as shown in Figure 6-15 (e) and (f).

TABLE I. Distortion parameters for left and right cameras.

	k_1	k_2	p_1	p_2	k_3
Left camera	-0.146	0.928	0.0085	-0.0023	0
Right camera	-0.1304	0.2702	-0.0089	0.0027	0

TABLE II. Intrinsic parameters for left and right cameras.

	f_x	f_y	α	c_x	c_y
Left camera	1 060.76	1 060.76	0	319.5	239.5
Right camera	1 011.102	1 011.102	0	319.5	239.5

The image-processing algorithm model will be introduced in Part II of this paper. The reconstructive accuracy of the 3-D coordinates, corresponding to the red markers attached to the catheter, directly affects the accuracy and effectiveness of the operation procedure. Two experiments were carried out to demonstrate the performance of the proposed system.

During the experimental operations, the master system was used to drive the slave system to insert or rotate the catheter inside the tube representing a blood vessel (Figure 6-15). Three frames were captured for results analysis.

First, the insertion/rotation operation was first carried out in a straight tube. Figure 6-16 (a) and (b) show the rectified results from the original images captured simultaneously by the left and right cameras during the procedure. Figure 6-16 (c) and (d) show the corresponding coordinates of the red markers and the reconstruction results, respectively, in the virtual reality environment.

The second frame is shown in Figure 6-15 (c) and 6-15 (d), which were explained in the previous context. The results of the last frame are shown in Figure 6-16.

Additionally, experiments were performed based on a bent pipe with different distances between the cameras and the tube, and the same intrinsic and extrinsic matrixes for the straight tube experiments are used. Figure 6-17 shows the first frame after an insertion operation, which was followed by a rotation operation, shown in Figure 6-18.

As we mentioned above, the actual distance between adjacent red markers was fixed (10 mm). Figure 6-19 provides the distances calculated by the 3-D coordinates of the two red markers attached to the tip of the catheter, over 10 frames. In Figure 6-19, the red bars show the actual distance, 10 mm, and the blue bars represent the calculated distance based on the proposed image-processing algorithm. The maximum error for the two red markers on the tip of the catheter was 0.93 mm less than 1 mm in these 10 frames.

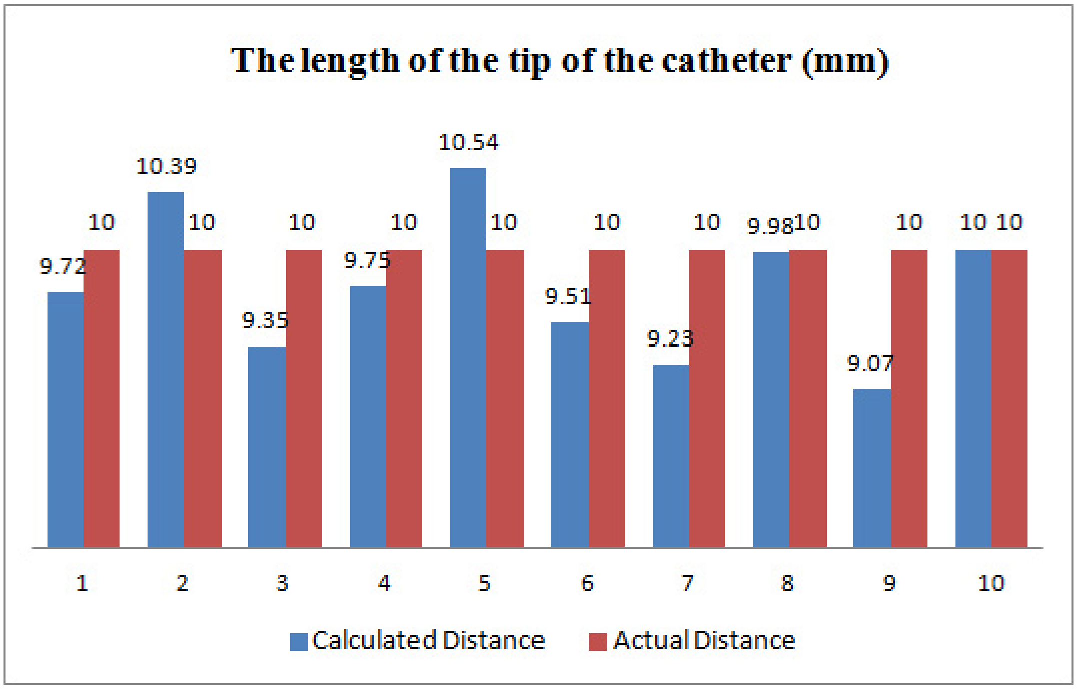


Figure 6- 19 The actual distance and calculated distance

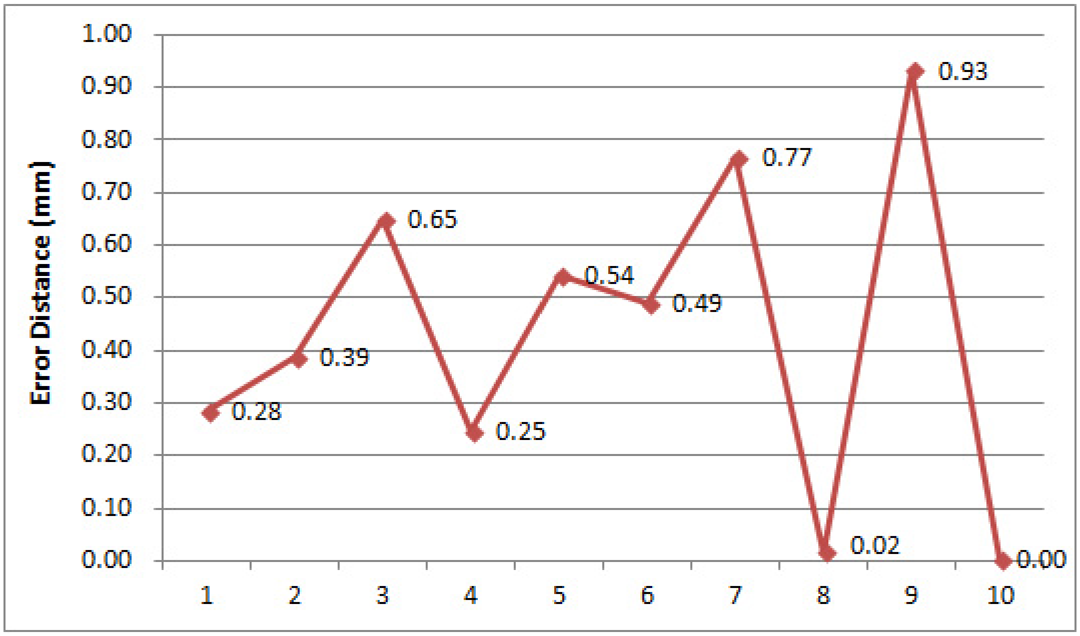


Figure 6- 20 The error distance for the catheter tip

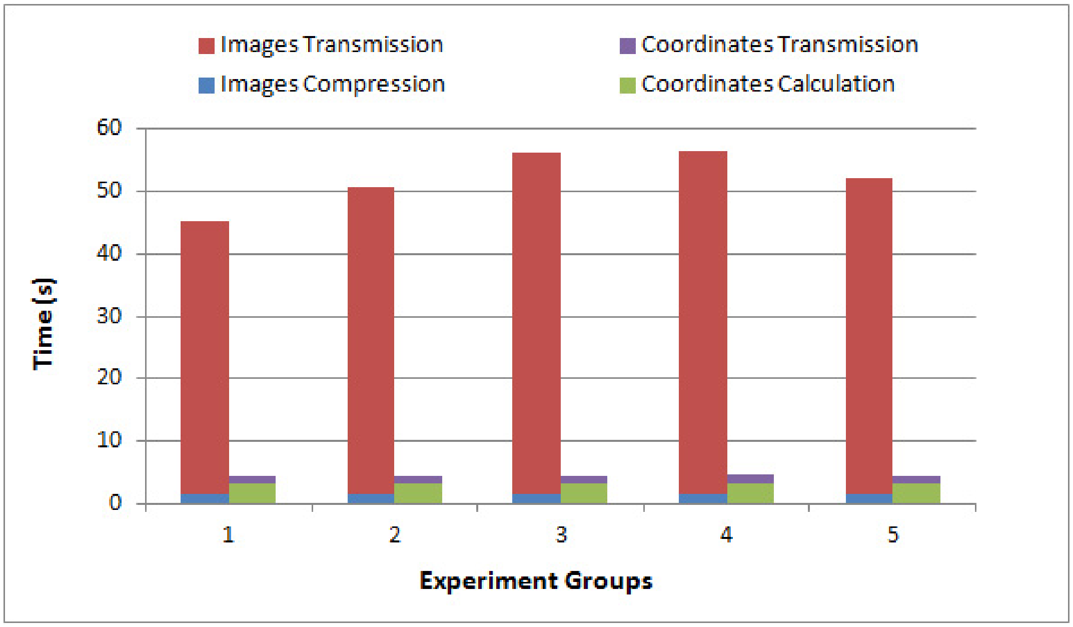


Figure 6- 21 The remote transmission experiments from Takamatsu to Beijing

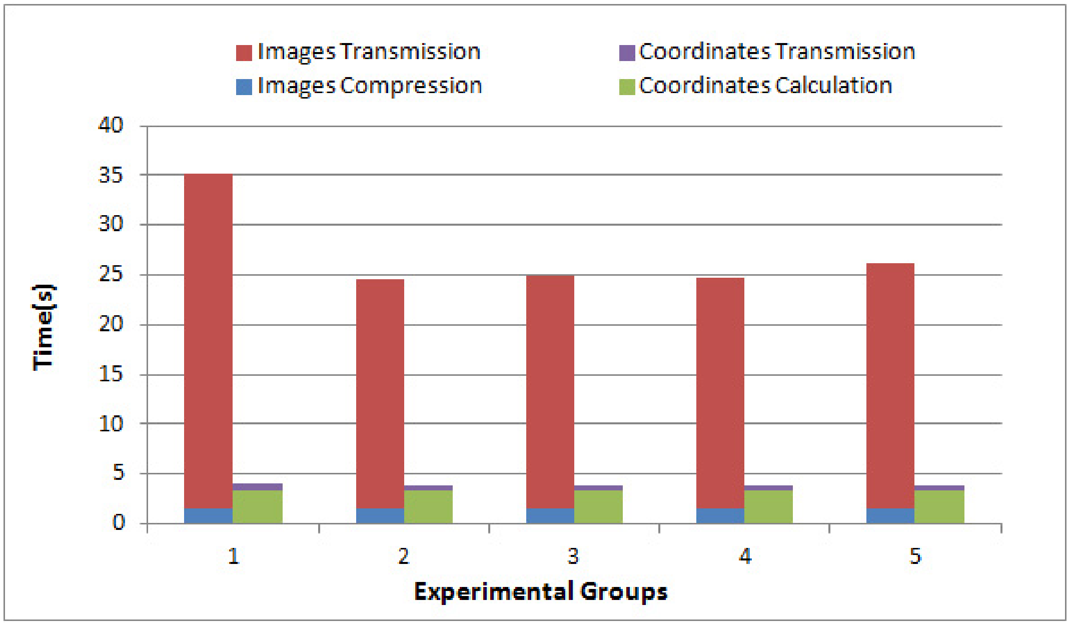


Figure 6- 22 The remote transmission experiments from Takamatsu to Harbin

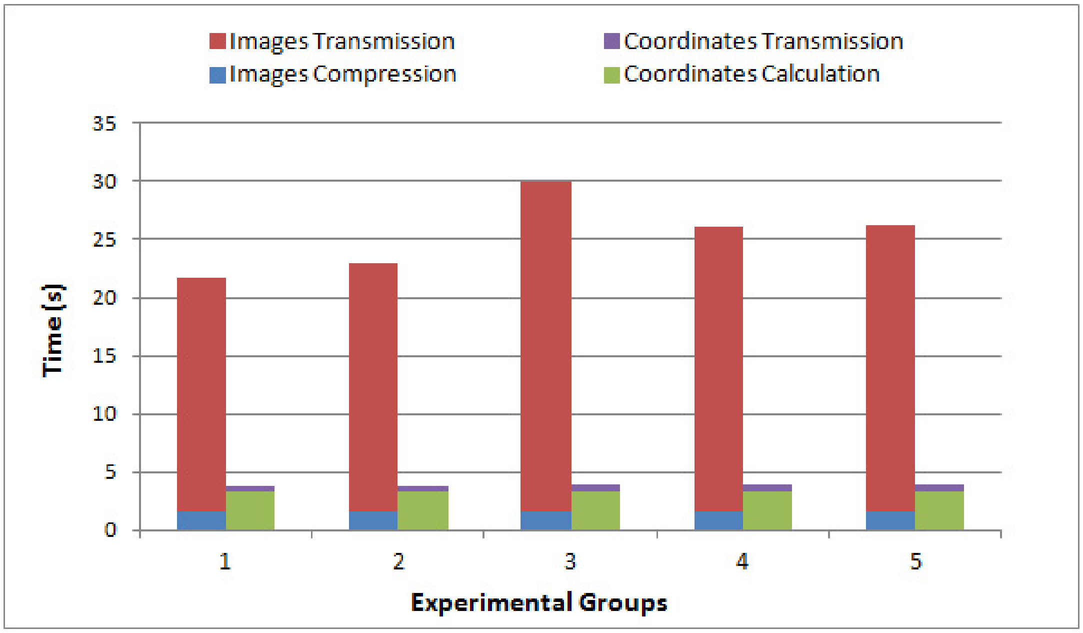


Figure 6- 23 The remote transmission experiments from Takamatsu to Shanghai

Based on our method, the visual feedback was converted from video images into 3-D coordinates, and then reconstructed in a virtual reality environment. Compared with conventional visual feedback, we were able to zoom in, zoom out, and rotate the virtual catheter and virtual tube (representing a blood vessel) to see more details. Furthermore, it can also be used as an assistant visual feedback if the surgeons are accustomed to the conventional image feedback. The proposed approach was quite effective for training operations.

Additionally, five experiments were conducted to compare the transmission time between China and Japan, before and after applying the proposed architecture. I delivered 105 pieces of images captured at the patient’s location and their corresponding red marker coordinates, based on our method used in the Takamatsu–Beijing experiments. The sizes of the images and their coordinate files were 1,294,398 bytes after compression and 28,350 bytes, respectively. The images were compressed as the conventional visual feedback transmitted from the patient’s location to the operation location. The compression algorithm based on JEPG encoding removes the redundant image data and can obtain a clear image with an appropriate compression ratio. The average compression time for each image is 15 milliseconds. The left bars in Figure 6-21 show the total transmission times for conventional visual feedback including images compression time (blue bars) and images transmission time (red bars); the right bars represent the sum of the time required for coordinate extraction (green bars), transmission (purple bars), and reconstruction in the virtual reality environment. The average time of coordinates’ calculation for two images (a left image and its right corresponding image) is 31 milliseconds and the time of reconstruction in the virtual reality environment is too short to be not counted. The images and their corresponding coordinates were transmitted in the same network condition. The transmission time was reduced considerably, providing better visual feedback.

Two additional remote transmission experiments were conducted between Takamatsu, Japan and Harbin and Shanghai, China, which have the different network conditions from Takamatsu to Beijing, to provide a comparison of the before and after results after applying the proposed method. I also delivered 105 pieces of images and the results are shown in Figure 6-22 and Figure 6-23.

6.3 Discussion

Obviously, shorter time delays for the transmission of visual feedback in surgery and a more stabilized network are desirable for the safety of the patient and surgeon, as well as for better operability during the procedure. However, it is impractical and expensive to locate dedicated fiber-optic lines worldwide for remote surgical procedures. The newly developed method described in this paper, which uses the conventional Internet infrastructure, may play an important role in the future use of remote surgical applications worldwide.

In terms of the feasibility of this approach for use in actual surgeries, the patient's blood vessels can be re-constructed in virtual reality environments in advance; many researchers have focused on this topic. The virtual blood vessel model customizes its size and shape to the vascular structure of the patient. At the patient's location, the catheter used in the surgery can be marked in a special way, allowing it to be detected and measured with radiological equipment. Several research groups have focused on a family of passive echogenic markers, in which the position and orientation of the surgical instrument can be determined in a 3-D ultrasound volume, based on an image-processing algorithm. They attached markers to the distal end of the instrument; the markers were detected and measured by ultrasound imaging, without the need for external tracking devices. For our proposed method, two radiological systems were required to detect the markers attached to the catheter. The 3-D coordinates were calculated, based on a specific image-processing algorithm, which converted the 3-D coordinates into a WCS for visual feedback. The 3-D coordinates were transmitted to the operation location and reconstructed in the virtual blood vessel model of the patient. Physicians were able to zoom in, zoom out, or rotate the patient's blood vessel model to see more detail of the surgical procedure. Additionally, several researchers have focused on the position and orientation measurement of a catheter with an electromagnetic sensor attached at the tip of the catheter. The electromagnetic sensor can provide the position and orientation of the tip of a catheter in the two references. The sensors may be attached to the catheter to provide the shape information not just the tip information in actual surgeries if the sensors do not alter the properties of a catheter and enlarge the size.

Due to physiological movements and other conditions, the real blood vessels of the patient tend to be different with the virtual vascular model in the VR system because the virtual model is built in advance. Researchers in [Wang11] used a rotational C-arm-based DSA machine to reconstruct 3D vascular structure. The DSA machine was calibrated by a special double-layered aluminum calibration phantom in order to obtain internal and external parameters at two predefined positions. After calibration, the DSA machine captured two rectified images at the two predefined positions. And the 3D position of the vascular structure was calculated by the two images. Depending on their research achievement, our proposed method may be used in a real surgery if two DSA machines are employed. The two DSA machines are firstly calibrated based on the double-layered aluminum calibration phantom separately. The vascular structure can be rebuilt. Additionally, the 3D position of the catheter with special makers can be reconstructed based on epipolar geometric constraints of the two images. The 3D positions of the moving parts of the blood vessels and the catheter will be sent to the virtual reality environment at the operation location. The vascular model in the VR system will be updated according to the 3D coordinates of the moving parts of the real blood vessels. And the catheter model will be reconstructed in the VR system based on the 3D coordinates of the real catheter.

6.4 Summary

In this part, I proposed a two-camera system to detect the position of a catheter inside a tube (used to represent a blood vessel), and transformed its red markers into 3-D coordinates using an image-processing algorithm at the patient’s location. The 3-D coordinates were then transmitted to the operation location and reconstructed in a virtual reality environment as visual feedback. The transmission time of visual feedback was effectively reduced due to the significant reduction in data volume using coordinates instead of actual images.

The proposed Internet-based novel catheter operating system was based on a master system and a slave system, which was explained in detail. We performed three experiments involving remote transmission of visual feedback between China and Japan, and compared the visual feedback transmission time before and after using the proposed method. The results were positive and demonstrate the feasibility of remote surgery using conventional network infrastructures.

In the future, we plan to improve the error of the catheter's tips in Figure 6-19 and Figure 6-20 by the sub-pixel interpolation adapted to each stereo-algorithm firstly, especially the sub-pixel level disparity calculation. Additionally, new detection algorithm and reconstruction methods for radiological equipment for our Internet-based will be focused on. Furthermore, remote animal experiments will be conducted based on the proposed method. This would allow us to determine the feasibility of our time-delay reduction approach in a real tele-surgery.

Chapter 7

Conclusions and future work

7.1 Conclusions

This thesis presents the reconstruction algorithms of realistic three-dimensional vascular models based on CT or MR images (Chapter 3), the development of a new master controller for the VR based training system (Chapter 5), and the design and implementation of a new prototype to reduce the transmission time of visual feedback based on VR-based method (Chapter 7).

Core components of virtual-reality surgical simulators and training systems are the realistic vascular models that are virtual representations of real blood vessels that display accurate displacement due to the fact that models with incorrect material properties and shapes could result in adverse training effects. In this study, two kinds of reconstruction algorithms were proposed. One is to rebuild the vascular structure based on normal and advanced CT or MR images. The normal images were employed to subtract their corresponding advanced images to remove human bones and other soft tissues, thereby remaining the data of blood vessels. This approach is adaptable to calculate the whole vascular structure. The other one is reconstruct the 3D vascular models based on advanced CT or MR images only according to marching cube and level sets algorithms. This approach is preferable to build a vascular segment due to its flexibility.

Based on analysis of requirements and state of the art in computer-assisted and robot-assisted training systems for vascular interventional surgery, a new master controller capable of contactless measuring the axial push-pull and twisting motions of a catheter with force feedback to the operators was proposed, whilst still preserving the natural ergonomic skills of conventional catheterization. A camera was employed to detect the motions of the catheter freely and operators can manipulate the real catheter intuitively with a familiar, ergonomic setting. Force feedback to the operator, in radial and axial directions, were generated by Ampere forces based on the calculated forces between the vascular model and the virtual catheter in a virtual reality environment or the forces from the sensing unit of the slave system. Several experiments were conducted to evaluate the performance of the new master controller. The experimental results show that the errors of the catheter sensing unit and haptic unit of the new master controller are in an acceptable range and it can be used as a controller in a computer-assisted or robot-assisted training system.

A newly developed method was described, based on virtual reality and image-processing technology, for efficiently reducing visual feedback transmission times, making it easier to control the time difference between haptic signals, control data, and image information.

This was achieved by reducing the amount of communication data in the image information. Several remote visual feedback transmission experiments were conducted to demonstrate the feasibility of the proposed prototype, via a cooperative surgical effort between research groups in China and Japan. The purpose of the experiments was to evaluate the performance of the new catheter operating system and its associated transmission times for visual feedback, by comparing performance before and after the application of the proposed architecture. The maximum error for catheter reconstruction on the tip of the catheter was 0.93 mm less than 1 mm in these 10 frames. Remote transmission experiments show that the transmission time of the visual feedback is reduced significantly based on the newly developed method.

The objective of my research is to present virtual-reality simulators based on a newly developed catheter operating system for training surgeons in minimally invasive surgery. These simulators can generate the realistic virtual-reality environment of blood vessels according to a patient’s special computed tomography (CT) or magnetic resonance imaging (MRI). In addition, the new master controller can allow simulating a surgeon’s operating skills to insert and rotate catheter as surgeon operates catheter directly and carry out interventions with haptic interfaces with force feedback, which provides the surgeon with a sense of resistance.

1) To reconstruct a realistic three-dimensional vascular model, a

median filter algorithm was employed to reduce the noise of CT images. Next is to rebuild the 3D surface of a vascular segment from CT or MR images using level sets. Finally, centerlines of a vascular segment were computed and the surface of a vascular segment was smoothed.

2) To maintain the natural endovascular catheterization skills of the interventional radiologist, a camera is employed to detect the axial and radial motions of a catheter simultaneously with the ability to allow interventional radiologists to use their dexterous skills while performing catheter-based interventions. Additionally, a haptic unit is developed to provide the interventionalists with force and torque feedback.

3) The transmission time of visual feedback was effectively reduced due to the significant reduction in data volume using coordinates instead of actual images.

7.2 Future work

In the future, I will realize the force feedback in the virtual reality environment including the tip feedback, generated from the tip contact between the catheter tip and the blood vessel wall, and the proximal feedback which is transmitted to the haptic unit of the master controller to provide operators with a resistance feeling and we plan to improve the error of the catheter's tips by the sub-pixel interpolation adapted to each stereo-algorithm firstly, especially the sub-pixel level disparity calculation. Additionally, new detection algorithm and reconstruction methods for radiological equipment for our Internet-based will be focused on. Furthermore, remote animal experiments will be conducted based on the proposed method. This would allow us to determine the feasibility of our time-delay reduction approach in a real tele-surgery.

References

[Padalino13] David J Padalino, Amit Singla, Walter Jacobsen and Eric M Deshaies, Enterprise Stent for Waffle-cone Stent-assisted Coil Embolization of Large Wide-necked Arterial Bifurcation Aneurysms, Surgical Neurology International, 2013.

[Spiotta12] Alejandro M Spiotta, Anne Marie Wheeler, Saksith Smithason, Ferdinand Hui and Shaye Moskowitz, Comparison of Techniques for Stent Assisted Coil Embolization of Aneurysms, Journal of Neurointerventional Surgery, 4(5):339-344, 2012.

[Basdogan01] C. Basdogan, C. Ho, and M.A. Srinivasan, "Virtual Environments for Medical Training: Graphical and Haptic Simulation of Common Bile Duct Exploration," IEEE ASME Trans. Mechatronics, vol. 6, no. 3, 2001, pp. 267-285.

[Çakmak00] H.K. Çakmak and U. Kühnapfel, Animation and Simulation Techniques for VR Training Systems in Endoscopic Surgery, Proc. Eurographics Workshop Animation and Simulation (EGCAS 2000), 2000, pp. 173-185.

[Brown02] J. Brown et al., Algorithmic Tools for Real-Time Microsurgery Simulation, Medical Image Analysis, vol. 6, no. 3, 2002, pp. 289-300.

[Tsang08] J.S Tsang, P.A. Naughton, S. Leong, A.D.K. Hill, C.J.

Kelly, A.L. Leahy, Virtual reality simulation in endovascular surgical training, The Surgeon, Volume 6, Issue 4, August 2008, pp. 214-220.

[Wei09] Daming Wei, Shin Hasegawa, Kunio Takahashi, Elena Ryzhii, and Xin Zhu, A Virtual Reality for Catheter-based EPS based on Whole-heart Model, International Journal of Bioelectromagnetism, Vol. 11, No. 1, 2009, pp.2-6.

[Basdogan07] Cagatay Basdogan, Mert Sedef, Matthias Harders and Stefan Wesarg, VR-Based Simulators for Training in Minimally Invasive Surgery, IEEE Computer Graphics and Application, Volume 27, No. 2, April 2007, pp. 54-66.

[Chiang11] Patricia Chiang, Yiyu Cai, Koon Hou Mak, Ei Mon Soe, Chee Kong Chui and Jianmin Zheng, A geometric approach to the modeling of the catheter-heart interaction for VR simulation of intra-cardiac intervention, Computers & Graphics, Volume 35, Issue 5, October 2011, pp. 1013-1022.

[Takashima07] Kazuo Takashima, Rei Shimomura, Takayuki Kitou, Hiroki Terada, Kiyoshi Yoshinaka and Ken Ikeuchi, Contact and friction between catheter and blood vessel, Tribology International, Volume 40, Issue 2, February 2007, pp. 319-328.

[Meier01] U. Meier, F.J. Garcia, N.C. Parr, C. Monserrat, J.A.Gil, V.Grau, M.C. Juan, M. Alcaniz, 3D surgery trainer with force feedback in minimally invasive surgery. International Congress Series, Volume 1230, June 2001, pp 32-37.

[Aggarwal06] R. Aggarwal, S.A. Black, J.R. Hance, A. Darzi and N.J.W. Cheshire, Virtual Reality Simulation Training can Improve Inexperimenced Surgeons' Endovascular Skills. European Journal of Vascular and Endovascular Surgery, Volume 31, Issue 6, June 2006, pp. 588-593.

[Neequaye07] Simon K. Neequaye, Rajesh Aggarwal, Isabelle Van Herzeele, Ara Darzi and Nicholas J. Cheshire, Endovascular skills training and assessment, Journal of Vascular Surgery, Volume 46, Issue 5, November 2007, pp. 1055-1064.

[Basdogan04] Cagatay Basdogan, Suvranu De, Jung Kim, Manivannan Muniyandi, Hyun Kim and Mandayam A, Srinivasan. Haptics in minimally invasive surgical simulation and training, Computer Graphics and Applications, IEEE, Volume 24, Issue 2, April 2004, pp. 56-64.

[Guiatni13] Mohamed Guiatni, Vincent Riboulet, Christian Duriez, Abderrahmane Kheddar, Stephane Cotin, "A Combined Force and Thermal Feedback Interface for Minimally Invasive Procedures Simulation", IEEE/ASME Transactions on Mechatronics, Vol. 18, No.3, pp. 1170-1181, 2013.

[Wang11] Junchen Wang, Takashi Ohya, Hongen Liao, Ichiro

Sakuma, Tianmiao Wang, Iwai Tohnai, Toshinori Iwai, "Intravascular catheter navigation using path planning and virtual visual feedback for oral cancer treatment", The International Journal of Medical Robotics and Computer Assisted Surgery, Vol. 7, pp. 214-224, 2011.

[Srimathveeravalli10] G. Srimathveeravalli, T. Kesavadas, X.Y. Li, "Design and fabrication of a robotic mechanism for remote steering and positioning of interventional devices", The International Journal of Medical Robotics and Computer Assisted Surgery, Vol. 6, pp.160-170, 2010.

[VIST, Mentice AB] http://www.mentice.com/our-simulators/

[Aggarwal06] R. Aggarwal, S.A. Black, J.R. Hance, A. Darzi, N.J.W. Cheshire, "Virtual Reality Simulation Training can Improve Inexperienced Surgeons' Endovascular Skills", European Journal of Vascular and Endovascular Surgery, Vol. 31, No. 6, pp. 588-593, 2006.

[Simbionix, Ltd.] http://simbionix.com/simulators/angio-mentor/

[Benjamin14] Benjamin A. Eslahpazir, Jerry Goldstone, Matthew T. Allemang, John C. Wang, Vikram S. Kashyap, "Principal considerations for the contemporary high-fidelity endovascular simulator design used in training and evaluation", Journal of Vascular Surgery, Vol. 59, No. 4, pp. 1154-1162, 2014.

[Desser07] Terry S. Desser, “Simulation-Based Training: The Next Revolution in Radiology Education?”, Journal of the American College of Radiology, Vol. 4, No. 11, pp. 816-824, 2007.

[HansenMedical]http://www.hansenmedical.com/int/en/vascular/magellan-robotic-system/product-overview

[Saliba08] Walid Saliba, Vivek Y. Reddy, Oussama Wazni, Jennifer E. Cummings, et al, “Atrial Fibrillation Ablation Using a Robotic Catheter Remote Control System”, Journal of the American College of Cardiology, Vol. 51, No. 25, pp. 2407-2411, 2008.

[Stereotaxis] http://www.stereotaxis.com/index.php

[Chun08] K.R. Julian Chun, Boris Schmidt, Bülent Köktürk, Roland Tilz, Alexander Fürnkranz, Catheter Ablation – New Developments in Robotics, Herz, Vol. 33, pp. 586-589, 2008.

[Payne12] Christopher J. Payne, Hedyeh Rafii-Tari, Guang-Zhong Yang, “A Force Feedback System for Endovascular Catheterisation”, Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 1298-1304, October 7-12, Vilamoura, Algarve, Portugal, 2012.

[Antoniou11] George A. Antoniou, Celia V. Riga, Erik K. Mayer, Nicholas J. W. Cheshire, Colin D. Bicknell, “Clinical applications of robotic technology in vascular and endovascular surgery”, Journal of Vascular Surgery, Vol. 53, No. 2, pp. 493-499, 2011.

[Das14] Paramita Das, Tarini Goyal, Andrew Xue, Sruti Kalatoor, Daniel Guillaume, “Simulation Training in Neurological Surgery”, Austin Journal of Neurosurgery, Vol. 1, No. 1, pp. 1-6, 2014.

[Binning12] Mandy Jo Binning, Adnan H. Siddiqui, Elad I. Levy, Leo Nelson Hopkins, Avoiding Complications During Percutaneous Cardiovascular Interventions: What Should We Learn From the Aviation Industry, Complications of Interventional Cardiovascular Procedures: A Case-Based Atlas, pp. 1-7, 2012.

[Cercenelli07] Laura Cercenelli, Emanuela Marcelli, Gianni Plicchi, “Initial Experience With a Telerobotic System to Remotely Navigate and Automatically Reposition Standard Steerable EP Catheters”, American Society for Artificial Internal Organs Journal, Vol. 53, No. 5, pp. 523-529, 2007.

[Marcelli08] Emanuela Marcelli, Laura Cercenelli, Gianni Plicchi, A Novel Telerobotic System to Remotely Navigate Standard Electrophysiology Catheters, Computers in Cardiology, 2008, pp. 137-140, September 14-17, Bologna, Italy, 2008.

[Tanimoto97] M. Tanimoto, F. Arai, T. Fukuda, H. Iwata, K. Itoigawa, Y. Gotoh, M. Hashimoto, M. Negoro, “Micro force sensor for intravascular neurosurgery”, Proceeding of the IEEE International Conference on Robotics and Automation, pp. 1561-1566, April 20-25, Albuquerque, NM, 1997.

[Negoro02] M. Negoro, M. Tanimoto, F. Arai, T. Fukuda, K. Fukasaku, I. Takahashi, S. Miyachi, “An Intelligent Catheter System Robotic Controlled Catheter System”, Interventional Neuroradiology, Vol. 7, pp. 111-113, 2002.

[Tercero12] C. Tercero, S. Ikeda, T. Fukuda, F. Arai, M. Negoro, I. Takahashi, “Numerical Comparison of Catheter Insertion Trajectory within Blood Vessel Model using Image Processing”, 2010 International Symposium on Micro-NanoMechatronics and Human Science, pp. 378-383, November 7-10, Nagoya, Japan, 2012.

[Guo12] Jian Guo, Shuxiang Guo, Nan Xiao, Xu Ma, Shunichi Yoshida, Takashi Tamiya, Masahiko Kawanichi, “A novel robotic catheter system with force and visual feedback for vascular interventional surgery”, International Journal of Mechatronics and Automation, Vol. 2, No. 1, pp. 15-24, 2012.

[Thakui09] Yogesh Thakui, Jeffrey S. Bax, David W. Holdsworth and Maria Drangova, “Design and Performance Evaluation of a Remote Catheter Navigation System”, IEEE Transactions on biomedical engineering, Vol.56, No. 7, pp. 1901-1908, 2009.

[Ma13] Xu Ma, Shuxiang Guo, Nan Xiao, Shunichi Yoshida, Takashi Tamiya, “Evaluating performance of a novel developed robotic catheter manipulating system”, Journal of Micro-Bio Robotics, Vol. 8, No. 3-4, pp. 133-143, 2013.

[Wei09] Daming Wei, Shin Hasegawa, Kunio Takahashi, Elena Ryzhii, Xin Zhu, “A Virtual Reality for Catheter-based EPS based on Whole-heart Model”, International Journal of Bioelectromagnetism, Vol. 11, No. 1, pp. 2-6, 2009.

[Lu11] Wang-sheng Lu, Wu-yi Xu, Jing Zhang, Da Liu, Da-min Wang, Po Jia, Zhi-cao Li, Tian-miao Wang, Da-peng Zhang, Zeng-min TIan, Yanjun Zeng, “Application study of medical robots in vascular intervention”, Vol. 7, No. 3, pp. 361-366, 2011.

[Kaehler08] Adrian Kaehler, Gary Bradski, “Learning OpenCV”, Publisher: O'Reilly Media, 2008.

[Zhang02] Zhengyou Zhang, "A Flexible New Technique for Camera Calibration", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 22(11), pp. 1330-1334, 2002.

[Guo14] Jin Guo, Shuxiang Guo, Takashi Tamiya, Hideyuki Hirata, Hidenori Ishihara, " A Virtual Reality-based Method of Decreasing Transmission Time of Visual Feedback for a Tele-operative Robotic Catheter Operating System ", The International Journal of Medical Robotics and Computer Assisted Surgery, to be published.

[Wang10] Tianmiao Wang, Dapeng Zhang, Liu Da. Remote-controlled vascular interventional surgery robot. The International Journal of Medical Robotics and Computer Assisted Surgery. 2010; 6: 194-201.

[Lumbar Fusion Surgery] Washington D.C.: Lumbar Fusion Surgery. http://www.gwhospital.com/hospital-services/the-gw-spine-center/lumbar-fusion-surgery

[Anvari07] Mehran Anvari. Telesurgery: Remote Knowledge Translation in Clinical Surgery. World Journal of Surgery. 2007; 31(8): 1545-1550.

[Mitsuishi07] Mamoru Mitsuishi. Medical Robot and Master Slave System for Minimally Invasive Surgery. Proceedings of the 2007 IEEE/ICME International Conference on Complex Medical Engineering; 2007 May 23-27; Beijing, China. p. 8-13.

[Guo12] Jin Guo, Shuxiang Guo, Nan Xiao. A Method of Decreasing Time Delay for A Tele-surgery System. Proceedings of 2012 IEEE International Conference on Mechatronics and Automation; 2012 August 5-8; Chengdu, China. p. 1191-1195.

[Liu11] Bo Liu, Xiangzhi Bai, Cai Meng, Shaojie Lv, Fugen Zhou. Image-guided Navigation System for Robot-based Vessel Intervention Surgery Based on Conventional Single Plane C-Arm. 2011 International Conference on Environment Science and Biotechnology; 2011 November 25-27; Maldives. p. 276-283.

[Arata07] Jumpei Arata, Hiroki Takahashi, Phongsaen Pitakwatchara. A remote surgery experiment between Japan and Thailand over Internet using a low latency CODEC system. Proceedings of 2007

IEEE International Conference on Robotics and Automation; 2007 April 10-14; Roma, Italy. p. 953-959.

[Arata06] Jumpei Arata, Hiroki Takahashi, Phongsaen Pitakwatchara. A remote surgery experiment between Japan-Korea using the minimally invasive surgical system. Proceedings of 2006 IEEE International Conference on Robotics and Automation; 2006 May 15-19; Orlando, Florida. p. 257-262.

[Marescaux01] J. Marescaux, J. Leroy, M. Gagner, F. Rubino, D. Mutter, M. Vix, S.E. Butner and M. K. Smith. Transatlantic robot-assisted telesurgery. NATURE. 2001; 413(27): 379-380.

[Zhou07] Jilin Zhou, Xiaojun Shen, Abdulmotaleb El Saddik, Nicolas D. Georganas. Design and Implementation of Haptic Tele-mentoring over the Internet. Proceedings of the 11th IEEE International Symposium on Distributed Simulation and Real-Time Applications; 2007 October 22-24; Washington, DC, USA. p. 201-208.

[Al-Mouhamed05] Mayez A. Al-Mouhamed, Onur Toker, Asif Iqbal, Syed M.S. Islam. Evaluation of Real-Time Delays for Networked Telerobotics. Proceeding of 2005 3rd IEEE International Conference on Industrial Informatics; 2005 August 10-12; Perth, Australia. p. 351-356.

[Islam11] Shafiqul Islam, Peter X. Liu, Abdulmotaleb El Saddik, Shichao Liu. Internet-based Teleoperation Systems with Time Varying Communication Delay. Instrumentation and Measurement Technology Conference (12MTC); 2011 May 10-12; Hangzhou, China. p. 1-6.

[Xiao12] Nan Xiao, Shuxiang Guo, Baofeng Gao, Xu Ma. Internet-based Robotic Catheter Surgery System. Proceeding of the IEEE International Conference on Automation and Logistics; 2012 August 15-17; Zhengzhou, China. p. 645-651.

[Kim10] Ki-Young Kim, Ho-Seok Song, Jung-Wook Suh, Jung-Ju Lee. A Teleoperated Minimally Invasive Surgical System with an Additional Degree of Freedom Manipulator. 2010 Fourth International Conference on Sensor Technologies and Applications; 2010 July 18-25; Venice, Italy. p.90-94.

[Xie07] Xie Xiaohui, Du Ruxu, Sun Lining, Du Zhijiang. Internet based telesurgery with a bone-setting system. Proceedings of the 2007 IEEE International Conference on Integration Technology; 2007 March 20-24; Shenzhen, China. p. 218-222.

[Smith10] Christian Smith, Patric Jensfelt. A Predictor for Operator Input for Time-Delayed Teleoperation, Mechatronics. Design Control Methodology. 2010; 20(7): 778-786.

[Zhang02] Zhengyou Zhang. A Flexible New Technique for Camera Calibration. IEEE Transactions on Pattern Analysis and Machine

Intelligence. 2002; 22(11): 1330-1334.

[Jayender08] JagaDeesan Jayender, Mahdi Azizian, Rajni V. Patel. Autonomous Image-Guided Robot-Assisted Active Catheter Insertion. IEEE Transactions on Robotics. 2008; 24(4): 858-871.

[Ma12] Xu Ma, Shuxiang Guo, Nan Xiao, Jian Guo. Development of a Novel Robotic Catheter Manipulating System with Fuzzy PID Control. International Journal of Intelligent Mechatronics and Robotics (IJIMR). 2012; 2(2): 58-77.

[Fu11] Yili Fu, Anzhu Gao, Hao Liu, and Shuxiang Guo. The master-slave catheterisation system for positioning the steerable catheter. International Journal of Mechatronics and Automation. 2011; 1: 143-152.

[Natarajan08] Sriram Natarajan, Aura Ganz. Efficient force feedback transmission system for tele surgery. 30th Annual International Conference of the IEEE Engineering in Medicine and Biology Society; 2008 August 20-24; British Columbia, Canada. p. 3245-3248.

[Richert09] D. Richert, C.J.B. Macnab. Direct Adaptive Force Feedback for Haptic Control with Time Delay. 2009 IEEE Toronto International Conference on Science and Technology for Humanity (TIC-STH); 2009 September 26-27; Toronto, Canada. p. 893-897.

[Stoll11] Jeffrey Stoll, Hongliang Ren, Pierre E. Dupont. Passive

Markers for Tracking Surgical Instruments in Real-Time 3-D Ultrasound Imaging. IEEE Transactions on Medical Imaging; 31(3): 563-575.

[Kaehler08] Adrian Kaehler, Gary Bradski. Learning OpenCV. Projection and 3D Vision. Publisher: O'Reilly Media; 2008. p. 405-416.

[Solomon03] Stephen B. Solomon, Timm Dickfeld, Hugh Calkins. Real-Time Cardiac Catheter Navigation on Three-Dimensional CT Images. Journal of Interventional Cardiac Electrophysiology. 2003; 8(1): 27-36.

[Wood05] Bradford J. Wood, Hui Zhang, Amir Durrani, Neil Glossop, Sohan Ranjan, David Lindisch. Navigation with Electromagnetic Tracking for Interventional Radiology Procedures: A Feasibility Study. Journal of Vascular and Interventional Radiology. 2005; 16: 493-505.

[Zhou10] Zhou Zhengdong, Pascal Haigron, Vincent Guilloux and Antoine Lucas, Virtual Reality Based Three-Dimensional Guide Wire Propagation Simulation For Endovascular Intervention, Transactions of Nanjing University of Aeronautics & Astronautics, Volume 27, No.1, March 2010, pp.62-69.

[Ye09] Xiufen Ye, Bing Qiao, Shuxiang Guo, Qingchang Guo, Reasearch of tissue deformation in virtual surgery simulation, Journal of Computer Applications, Volume 29, Issue 2, February 2009, pp. 568-573.

[Zhang08] Zhang Qiukui, Pascal Haigron, Luo limin and Shu huazhong, FEM model for real-time guide wire simulation in vasculature, Journal of Southeast University, Volume 24, No.1, March 2008, pp. 50-54.

[Wang11] Xiaonan Wang, Max Meng, Perspective of Active Capsule Endoscope: Actuation and Localization, International Journal of Mechatronics and Automation, Vol.1, No.1, 2011, pp. 38-45.

[Abdulla11] Shahab Abdulla, Peng Wen, Robust Internal Model Control for Depth of Anaesthesia, International Journal of Mechatronics and Automation, Vol.1, No.1, 2011, pp. 1-8.

[Wu11] Y.C. Wu and J.S. Chen, Toward the Identification of EMG-Signal and Its Bio-Feedback Application, International Journal of Mechatronics and Automation, Vol.1, No.2, 2011, pp. 112-120.

[Szewczyk11] Jérôme Szewczyk, Emilie Marchandise, Patrice Flaud, Laurent Royon, and Raphaël Blanc, Active Catheters for Neuroradiology, Journal of Robotics and Mechatronics, Vol.23, No.1, pp. 105-115, 2011.

[Nakayama11] Masano Nakayama, Satoko Abiko, Xin Jiang, Atsushi Konno, Masaru Uchiyama, Stable Soft-Tissue Fracture Simulation for Surgery Simulator, Journal of Robotics and Mechatronics, Vol.23, No.4, pp. 589-597, 2011.

[Tomono09] Masahiro Tomono, 3D Object Modeling and Segmentation Using Image Edge Points in Cluttered Environments, Journal of Robotics and Mechatronics, Vol.21, No.6, pp. 672-679, 2009.

[Yamaguchi07] Yuzo Yamaguchi, Takumi Yoshimizu, Yoshihiro Muta, and Mitsuo Tashiro, A Study on Catheter Drives for Automatic Aspiration Working with Ventilator, Journal of Robotics and Mechatronics, Vol.19, No.3, pp. 331-338, 2007.

[Sato06] Daisuke Sato, Ryosuke Kobayashi, Akira Kobayashi, Shohei Fujino, and Masaru Uchiyama, Soft Tissue Pushing Operation Using a Haptic Interface for Simulation of Brain Tumor Resection, Journal of Robotics and Mechatronics, Vol.18, No.5, pp. 634-642, 2006.

[Luboz09] Vincent Luboz, Rafal Blazewski, Derek Gould, Fernando Bello, Real-time guidewire simulation in complex vascular models, Image and Vision Computing, Vol.25, pp. 827–834, 2009.

[Wu11] Xunlei Wu, Vincent Luboz, Karl Krissian, Stephane Cotin, Steve Dawson, Segmentation and reconstruction of vascular structures for 3D real-time simulation, Medical Image Analysis, Vol. 15, pp. 22-34, 2011.

[Newman06] Timothy S. Newman, Hong Yi, A survey of the marching cubes algorithm, Computers & Graphics, Vol. 30, pp. 854-879, 2006.

Publication List

International Journal Papers

1. **Jin Guo,** Shuxiang Guo, Takashi Tamiya, Hideyuki Hirata, Hidenori Ishihara, "A Virtual Reality-based Method of Decreasing Transmission Time of Visual Feedback for a Tele-operative Robotic Catheter Operating System", The International Journal of Medical Robotics and Computer Assisted Surgery (IJMRCAS), Accepted, In press, 2014.
2. **Jin Guo,** Shuxiang Guo, Nan Xiao, Baofeng Gao, "Virtual Reality Simulators based on a Novel Robotic Catheter Operating System for Training in Minimally Invasive Surgery", Journal of Robotics and Mechatronics, Vol. 24, No. 4, pp. 649-655, August, 2012.
3. **Jin Guo**, Shuxiang Guo, Takashi Tamiya, Hideyuki Hirata, Hidenori Ishihara, " Design and Performance Evaluation of a Master Controller for Endovascular Catheterization", International Journal of Medical Informatics, submitted, 2014.

International Conference Papers

4. **Jin Guo,** Shuxiang Guo, "A Haptic Interface Design for a VR-based Unskilled Doctor Training System in Vascular Interventional Surgery", Proceedings of 2014 IEEE International Conference on Mechatronics and Automation, pp.1259-1263, August, 2014.

5. **Jin Guo,** Shuxiang Guo, Nan Xiao, "An Image-processing based Prototype of Decreasing Transmission Time of Visual Information for a Tele-operative Surgical System", Proceedings of 24th 2013 International Symposium on Micro-NanoMechatronics and Human Science, pp.102-107, November, 2013.

6. **Jin Guo,** Shuxiang Guo, Nan Xiao, Thomas Dauteuille, "A VR-based Training System for Vascular Interventional Surgery", Proceedings of 2013 ICME International Conference on Complex Medical Engineering (ICME CME 2013), pp.575-579, May, 2013.

7. **Jin Guo,** Shuxiang Guo, Nan Xiao, " A Method of Decreasing Transmission Time of Visual Feedback for the Internet-based Surgical Training System ", Proceedings of 2013 IEEE International Conference on Mechatronics and Automation, pp.914-919, August, 2013.

8. **Jin Guo,** Shuxiang Guo, Nan Xiao, " A Method of Decreasing Time Delay for A Tele-surgery System", Proceedings of 2012 IEEE

International Conference on Mechatronics and Automation, pp.1191-1195, August, 2012.

9. Shuxiang Guo, Mohan Qu, Baofeng Gao, **Jin Guo**, " Deformation of the Catheter and 3D Blood Vessel Model for a VR-based Catheter System ", Proceedings of 2013 IEEE International Conference on Mechatronics and Automation, pp.861-866, August, 2013.

10. Xu Ma, Shuxiang Guo, Nan Xiao, Baofeng Gao, **Jin Guo**, Takashi Tamiya and Masahiko Kawanishi, " Remote Catheterization Using a New Robotic Catheter Manipulating System ", Proceedings of 2013 ICME International Conference on Complex Medical Engineering (ICME CME 2013), pp.394-398, May, 2013.

11. Baofeng Gao, Shuxiang Guo, Nan Xiao and **Jin Guo**, "Design of the Virtual Reality based Robotic Catheter System for Minimally Invasive Surgery Training", Proceedings of the 2012 IEEE International Conference on Automation and Logistics, pp.622-627, August, 2012.

12. Baofeng Gao, Shuxiang Guo, Nan Xiao and **Jin Guo**, "Development of a 3D blood vessel model for the Simulation of the Minimally Invasive Surgery", Proceedings of 2012 IEEE International Conference on Mechatronics and Automation, pp.1393-1398, August, 2012.

13. Xu Ma, Shuxiang Guo, Nan Xiao, **Jin Guo**, Shunichi Yoshida, Takashi Tamiya, Masahiko Kawanishi and Baofeng Gao, "NARX Model-based Identification for the Developed Novel Robotic Catheter Manipulating System", Proceedings of 2012 IEEE International Conference on Mechatronics and Automation, pp.2225-2229, August, 2012.

14. Baofeng Gao, Shuxiang Guo, Nan Xiao, **Jin Guo**, Xufeng Xiao, Shu Yang, Mohan Qu, "Construction of 3D Vessel Model of the VR Robotic Catheter System", Proceeding of 2012 IEEE International Conference on Information and Automation, pp. 783-788, June, 2012.

Appendix I

The program code of the robot-assisted catheter system (Visual Studio 2010)

```
#include "stdafx.h"
#include "MasterData.h"
#include "DlgMasterSever.h"
#include "MasterGlobalVariables.h"
#include "MasterGlobalFunctions.h"
#include "afxdialogex.h"
/*#include "serialport.h"*/

#define DEBUG_LAN_WORK 0

IMPLEMENT_DYNAMIC(CDlgMasterSever, CDialog)

CDlgMasterSever::CDlgMasterSever(CWnd* pParent /*=NULL*/)
		: CDialog(CDlgMasterSever::IDD, pParent)
		, m_bIsListening(false)
		, m_nPortNum(6803)
		, m_nportNO(6803)
		,m_bIsSerialPortOpen(false)
		,m_bIsSerialComm(false)
		,m_nSerialCNT(0)
		,m_bIsLanComm(false)
{
		//m_serversocket = new CServerSocket();
}

CDlgMasterSever::~CDlgMasterSever()
{
		//delete m_serversocket;
}

void CDlgMasterSever::DoDataExchange(CDataExchange* pDX)
{
		CDialog::DoDataExchange(pDX);
		DDX_Text(pDX, IDC_EDIT1, m_nportNO);
}


BEGIN_MESSAGE_MAP(CDlgMasterSever, CDialog)
		ON_BN_CLICKED(IDOK, &CDlgMasterSever::OnBnClickedBtnCreateServer)
		ON_BN_CLICKED(IDCANCEL, &CDlgMasterSever::OnBnClickedBtnSendByLan)
		ON_BN_CLICKED(IDC_BTN_EXIT, &CDlgMasterSever::OnBnClickedBtnExit)
		ON_MESSAGE(WM_MSG_LAN_RX, OnGetLanRXMsg)			//		for
lan rx
		ON_MESSAGE(WM_MSG_LAN_TX, OnGetLanTXMsg)			//
		for lan tx
		ON_MESSAGE(WM_COMM_RXCHAR,OnGetComRxMsg)		// for com rx
		ON_MESSAGE(WM_MSG_COM_TX, OnGetComTXMsg)		//for com tx
		ON_BN_CLICKED(IDC_BTN_STRCOM, &CDlgMasterSever::OnBnClickedBtnStrcom)
```

```
        ON_BN_CLICKED(IDC_BTN_OPENCOM, &CDlgMasterSever::OnBnClickedBtnOpencom)
END_MESSAGE_MAP()



void CDlgMasterSever::showMessage(const char* msg)
{
        CListBox* plistbox = (CListBox*)GetDlgItem(IDC_LIST1);
        plistbox->SetTopIndex(plistbox->AddString((LPCTSTR)msg));
}


void CDlgMasterSever::OnBnClickedBtnCreateServer()
{
        MasterSTFromCom[0] = 0xa;
        MasterSTFromCom[1] = 0xb;
        MasterSTFromCom[2] = 0xc;
        MasterSTFromCom[3] = 0xd;
        MasterSTFromCom[15] = 0x1;

        if (!m_bIsListening)
        {
                                UpdateData(true);
                                m_nPortNum = m_nportNO;
                                m_listensocket.m_hwnd = this->GetSafeHwnd();

                                AfxSocketInit();
                                if (m_listensocket.Create(m_nPortNum, SOCK_STREAM, 0))
                                {
                                                                showMessage("Server
built.");

                                                                BYTE    *p;
                                                                char    temp[100];
                                                                struct   hostent   *hp;
                                                                CString ip;

                                                                if(gethostname(temp,
sizeof(temp)) == 0)

        if((hp=gethostbyname(temp)) != 0)
                                                                {

        p    =(BYTE    *)hp-> h_addr;

        ip.Format(_T("Sever IP: %d.%d.%d.%d : %d"),    p[0],    p[1],    p[2],    p[3], m_nPortNum);

        showMessage((LPSTR)(LPCTSTR)ip);
                                                                }

                                                                if
( !m_listensocket.Listen(5))
                                                                {

        int nErrorCode = 0;
```

```
         nErrorCode = m_listensocket.GetLastError();

         TCHAR errMsg[2048] = {0};

         FormatMessage(    FORMAT_MESSAGE_FROM_SYSTEM,

                                                                    NULL,

                                                                    nErrorCode,

         MAKELANGID(LANG_NEUTRAL, SUBLANG_DEFAULT),

                                                                    errMsg,

         sizeof(errMsg)/sizeof(TCHAR),

                                                                    NULL
         );

         showMessage((LPSTR)errMsg);

         return;
                                                            }else
                                                            {
         m_bIsListening = !m_bIsListening;

         GetDlgItem(IDOK)->SetWindowText(_T("Disconnect"));
                                                            }

                          }else                      // create error
                          {
                                                     int nErrorCode = 0;
                                                     nErrorCode =
m_listensocket.GetLastError();
                                                     TCHAR errMsg[2048] = {0};
                                                     FormatMessage(
         FORMAT_MESSAGE_FROM_SYSTEM,

                                      NULL,

                                      nErrorCode,

                                      MAKELANGID(LANG_NEUTRAL,
SUBLANG_DEFAULT),

                                      errMsg,

                                      sizeof(errMsg)/sizeof(TCHAR),

                                      NULL           );

                                                     showMessage((const
char*)errMsg);
                                                        return;
                                    }
        }else                               //          is listening
        {
                                    if (m_listensocket.m_bIsConnected)
                                    {
        m_listensocket.m_SeverSocket->Send("sbs", 3, 0);
                                    }
                                    m_listensocket.Close();
                                    m_bIsListening = !m_bIsListening;
                                    GetDlgItem(IDOK)->SetWindowText(_T("Create Server"));
                                    showMessage("Server disconnected.");
        }

}

void CDlgMasterSever::OnBnClickedBtnSendByLan()                 // btn send by lan
{
                if (m_listensocket.m_bIsConnected)
                {
                                        m_serversocket = m_listensocket.m_SeverSocket;
                                        m_serversocket->Send(MasterSTFromCom, 16, 0);
                }
}

void CDlgMasterSever::OnBnClickedBtnExit()
{
        if (m_bIsListening)
        {
                m_listensocket.Close();
        }
        OnCancel();
}

LRESULT CDlgMasterSever::OnGetLanRXMsg(WPARAM wParam, LPARAM lParam)
{
                        switch(wParam)
                        {
                        case        SOCK_CLIENT_REQUEST :
                                                        {

        showMessage("Client Request.");
                                                                        break;
                                                        }
                        case SOCK_SERVER_RECIEVED:
                                                        {
                                                                //
        showMessage((const char*)recvbuf);
                                                                        for (int i = 0;
i<16; i++)
                                                                        {

        *(SlaveSTFromLan+i) = *(recvbuf+i);
                                                                        }
                                    break;
                            }
            }
            return 0;
}

LRESULT CDlgMasterSever::OnGetLanTXMsg(WPARAM wParam, LPARAM lParam)
{

    if (m_listensocket.m_bIsConnected)
    {
            m_serversocket = m_listensocket.m_SeverSocket;
            m_serversocket->Send(MasterSTFromCom, 16, 0);
            return 0;
    }
            return -1;
}

void CDlgMasterSever::OnBnClickedBtnStrcom()
{

            CButton* pBTN = (CButton*)GetDlgItem(IDC_BTN_STRCOM);

            if (!m_bIsListening)
            {
                    MessageBox(_T("No client connected or No server is
build."), _T("Error"), MB_OK|MB_ICONEXCLAMATION);
                    return;
            }
            if (!m_bIsSerialPortOpen)
            {
                    MessageBox(_T("No serial port is opened."), _T("Error"),
MB_OK|MB_ICONEXCLAMATION);
                    return;
            }

//      start serial port and Lan communication
            if (m_bIsLanComm)
            {
                    pBTN->SetWindowText(_T("Start Comm"));
//              stop serial port comm
                    m_bIsSerialComm = false;
                    m_SerialPort.StopMonitoring();
                    ThreadSerialPort->SuspendThread();

//              stop LAN comm
                    ThreadLanTX->SuspendThread();
                    m_bIsLanComm = false;
            }else
            {
                    pBTN->SetWindowText(_T("Stop Comm"));
//              start serial port comm
                    m_SerialPort.StartMonitoring();
                    m_bIsSerialComm = true;

                    ThreadSerialPort =
AfxBeginThread(SerialPortTXDataThread,(LPVOID)m_hDlgHwnd);

                                                //              start LAN comm
                                                ThreadLanTX = AfxBeginThread( LanTXDataThread,
(LPVOID)m_hDlgHwnd);
                                                m_bIsLanComm = true;
                            }

        }

void CDlgMasterSever::OnBnClickedBtnOpencom()
{
                                                CButton* pBTN =
(CButton*)GetDlgItem(IDC_BTN_OPENCOM);
                                        //      pBTN->EnableWindow(false);
                                                CComboBox* pCBX1 =
(CComboBox*)GetDlgItem(IDC_COMBO1);

                                                CComboBox* pCBX2 =
(CComboBox*)GetDlgItem(IDC_COMBO2);
                                                if (m_bIsSerialPortOpen)
                                                {
                                                        pBTN->SetWindowText(_T("Open
SerialPort"));

                                                        m_SerialPort.ClosePort();
                                                        m_bIsSerialPortOpen = true;
                                        //              MessageBox(_T("Serial Port has already
opened."), _T("ERROR"), MB_OK|MB_ICONERROR);

                                                        return;
                                                }

                                                pBTN->SetWindowText(_T("Close SerialPort"));
                                                char strTemp[32];
                                                pCBX1->GetLBText(pCBX1->GetCurSel(),
(LPTSTR)strTemp);

                                                m_nSerialPortNum = strTemp[3]-48;

                                                switch(pCBX2->GetCurSel())
                                                {
                                                case 0:
                                                          {     m_nSerialPortBuad = 4800; break;
        }
                                                case 1:
                                                          {     m_nSerialPortBuad = 9600; break;
        }
                                                case 2:
                                                          {     m_nSerialPortBuad = 19200;
break;}
                                                case 3:
                                                         {      m_nSerialPortBuad = 38400;
break;}
                                                }
                                                m_bIsSerialPortOpen = true;
                                                m_SerialPort.InitPort(this, m_nSerialPortNum,
m_nSerialPortBuad, 'N', 8, 1 );
}
```

```
BOOL CDlgMasterSever::OnInitDialog()
{
    CDialog::OnInitDialog();
    CComboBox* pCBX =(CComboBox*)GetDlgItem(IDC_COMBO2);
    pCBX->SetCurSel(2);
    findCommPort();

    m_hDlgHwnd = this->GetSafeHwnd();
    return TRUE;   // return TRUE unless you set the focus to a control
}

void CDlgMasterSever::findCommPort()
{
//  CStringArray strCommArr;
    HKEY hKey;
    int rtn;
    CComboBox* pCBX = (CComboBox*)GetDlgItem(IDC_COMBO1);

    rtn = RegOpenKeyEx( HKEY_LOCAL_MACHINE, _T("Hardware\\DeviceMap\\SerialComm"),
        NULL, KEY_READ, &hKey);
    if( rtn == ERROR_SUCCESS)
    {
                int   i=0;
                char   portName[256],commName[256];
                DWORD   dwLong,dwSize;
                while(1)
                {
                    dwSize =   sizeof(portName);
                    dwLong    =    dwSize;
                    rtn = RegEnumValue( hKey, i, portName, &dwLong,
                        NULL, NULL, (PUCHAR)commName,
&dwSize );
                    if( rtn == ERROR_NO_MORE_ITEMS )
                    break;
                    pCBX->AddString(_T(commName));
                        //      strCommArr.Add(commName);
                    i++;
                }
                RegCloseKey(hKey);
    }

    pCBX->SetCurSel(0);
}


LRESULT CDlgMasterSever::OnGetComRxMsg(WPARAM ch,LPARAM port)
{

    BYTE temp    = (BYTE)ch;

    if (temp == 0xAB)
    {
        g_MSTFromCom = 0;
        g_COM_Event.SetEvent();
        g_LAN_Event.SetEvent();
```

```
		}
		if (g_MSTFromCom > 16)
		{
			return -1;
		}
		MasterSTFromCom[g_MSTFromCom++] = temp;
		return 0;
}

LRESULT CDlgMasterSever::OnGetComTXMsg(WPARAM wParam, LPARAM lParam)
{
		m_SerialPort.WriteToPort(SlaveSTFromLan, 16);
		return 0;
}


#include "stdafx.h"
#include "MasterData.h"
#include "DlgDataReceive.h"
#include "MasterGlobalFunctions.h"
#include "Serialport.h"

#define DEBUG_WM 0
#define MSCOMM 0

IMPLEMENT_DYNAMIC(CDlgDataReceive, CDialog)

CDlgDataReceive::CDlgDataReceive(CWnd* pParent /*=NULL*/)
: CDialog(CDlgDataReceive::IDD, pParent)
,m_PortNum(5)
,m_IsPortCreat(0)
,m_IsPortOpen(0)
,m_InBufferSize(1024)
,m_OutBufferSize(1024)
{

}

CDlgDataReceive::~CDlgDataReceive()
{
		DataBaseQuit();
		gbSerialPortThreadBreak = FALSE;
}

void CDlgDataReceive::DoDataExchange(CDataExchange* pDX)
{
		CDialog::DoDataExchange(pDX);
}


BEGIN_MESSAGE_MAP(CDlgDataReceive, CDialog)
		ON_BN_CLICKED(IDOK, &CDlgDataReceive::OnBnClickedOk)
		ON_BN_CLICKED(IDC_BUTTON_START, &CDlgDataReceive::OnBnClickedButtonStart)
		ON_BN_CLICKED(IDC_BUTTON_STOP, &CDlgDataReceive::OnBnClickedButtonStop)
		ON_BN_CLICKED(IDCANCEL, &CDlgDataReceive::OnBnClickedCancel)
```

```
        ON_WM_PAINT()
        ON_WM_TIMER()
        ON_WM_DESTROY()
        ON_WM_CLOSE()
        ON_WM_CREATE()

END_MESSAGE_MAP()


void CDlgDataReceive::OnBnClickedOk()
{
//              int portNum;
//              portNum = m_PortNum;
//              SerialPortConnect(portNum);
}

void CDlgDataReceive::OnBnClickedButtonStart()
{
        UINT lpParam=0;
        AfxBeginThread(LPVOID lpParam)

        GetDlgItem(IDC_BUTTON_START)->EnableWindow(FALSE);
        GetDlgItem(IDC_BUTTON_STOP)->EnableWindow(TRUE);

        //-----------------------------------------------------------------------------------------------------------------------
----------------------
//      Serial Port
        SERIAL_INFO SerialInfo;
        SerialInfo.PortNum = m_PortNum;
//      ThreadSerialPort = AfxBeginThread(SerialPortTXDataThread,(LPVOID)&SerialInfo);
        gbSerialPortThreadBreak = FALSE;

//--------------------------------------------------------------------------------------------------------------------------------
---------
//      Draw
        gIsDrawStart = 1;
        gIsDrawCircle = 0;
        SetTimer(0,100,NULL);
//--------------------------------------------------------------------------------------------------------------------------------
---------
//      Data Save
        nExcelFileCnt = 1;
        DataBaseInit();
        gbSaveDataThreadBreak = FALSE;
        THREAD_INFO ThreadInfo;
        ThreadSaveData = AfxBeginThread(SaveDataThread, (LPVOID)&ThreadInfo);
}

void CDlgDataReceive::OnBnClickedButtonStop()
{
        gbSerialPortThreadBreak = TRUE;

        m_IsPortOpen = 0;
        GetDlgItem(IDC_BUTTON_START)->EnableWindow(TRUE);
        GetDlgItem(IDC_BUTTON_STOP)->EnableWindow(FALSE);
        KillTimer(0);
        gbSaveDataThreadBreak = TRUE;
}

void CDlgDataReceive::OnBnClickedCancel()
{
        OnCancel();
        //MessageBox(_T(""),_T(""),MB_OK);
}

void CDlgDataReceive::OnPaint()
{
        CPaintDC dc(this); // device context for painting

}

void CDlgDataReceive::OnTimer(UINT_PTR nIDEvent)
{
        switch(nIDEvent)
        {
        case 0:
                {
                        DrawWave();
                        break;
                }

        }

        CDialog::OnTimer(nIDEvent);
}


void CDlgDataReceive::DrawWave()
{
        CWnd *WaveWindowPICT;
        CRect WaveWindowRect;
        CDC *pDC;
        CBitmap memBitmap;
        CBitmap* pOldBmp = NULL;
        CPen *Oldpen;
        CFont   font,font2;
        CFont   *pOldFont=NULL;
        POINT EndPoint;
        double dCalibration;
        int nCntTmp;
        int nRow;

        //      InvalidateRect(WaveWindowRect);
        font.CreatePointFont(80,_T("Arial"));
        CPen GraphYellowPen(PS_SOLID, 3, RGB(255,255,0));
        CPen SolidGreenPen(PS_SOLID, 2, RGB(50, 150,50));
        CPen GraphPen(PS_SOLID, 1, RGB(0,0,0));

        WaveWindowPICT = GetDlgItem(IDC_STATIC);
        WaveWindowPICT->GetClientRect(WaveWindowRect);
        pDC = WaveWindowPICT->GetDC();
        WaveWindowPICT->Invalidate();
        // Create memory draw dc
        memDC.CreateCompatibleDC(pDC);
        memBitmap.CreateCompatibleBitmap(pDC, WaveWindowRect.right, WaveWindowRect.bottom);
        pOldBmp = memDC.SelectObject(&memBitmap);
        memDC.BitBlt(WaveWindowRect.left,WaveWindowRect.top,WaveWindowRect.right,WaveWindow
Rect.bottom, pDC, 0,0, SRCCOPY);

/*******************************************************************************************
***************/
        memDC.FillSolidRect(WaveWindowRect, (0,0,0));
        Oldpen = memDC.SelectObject(&SolidGreenPen);

        for (int i =1; i<40;i++)
        {
                memDC.MoveTo(WaveWindowRect.Width()*i/40, WaveWindowRect.top);
                memDC.LineTo(WaveWindowRect.Width()*i/40, WaveWindowRect.bottom);
        }
        for (int i =1; i<10; i++)
        {
                memDC.MoveTo(WaveWindowRect.left, WaveWindowRect.Height()*i/10);
                memDC.LineTo(WaveWindowRect.right, WaveWindowRect.Height()*i/10);
        }

        if (gIsDrawStart )
        {
                dCalibration =WaveWindowRect.Width()/256.0;
                if (gTotalCnt > DATANUM/2)
                {
                        gIsDrawCircle = 1;
                }

                if (!gIsDrawCircle)
                {
                        nCntTmp = (gTotalCnt<256)? gTotalCnt : 256;
                }
                else
                {
                        nCntTmp = 256;
                }
                Oldpen = memDC.SelectObject(&GraphYellowPen);
                memDC.MoveTo(WaveWindowRect.right,WaveWindowRect.bottom);

                for (int i=0;i<nCntTmp;i++)
                {
                        nRow = (gTotalCnt-i<0)?(DATANUM-i+gTotalCnt): (gTotalCnt-i);
                         if (gTotalCnt-i<0)
                         {
                                 EndPoint.x = 1;
                         }
                        EndPoint.x = WaveWindowRect.right - (i-1)*dCalibration;
                        EndPoint.y = WaveWindowRect.bottom -
(MasterStatus[nRow][LC]/4096.0)*(WaveWindowRect.Height());
                        memDC.LineTo(EndPoint);
                }
        }
```

```
	pDC->BitBlt(WaveWindowRect.left,WaveWindowRect.top,WaveWindowRect.right,
WaveWindowRect.bottom, &memDC, 0,0,SRCCOPY);
	memDC.SelectObject(pOldBmp);
	memDC.SelectObject(Oldpen);
	memDC.DeleteDC();
	memBitmap.DeleteObject();

	GraphYellowPen.DeleteObject();
	GraphPen.DeleteObject();
	SolidGreenPen.DeleteObject();
	font.DeleteObject();
	font2.DeleteObject();
	ReleaseDC(pDC);
	ReleaseDC(&memDC);
}

int CDlgDataReceive::OnCreate(LPCREATESTRUCT lpCreateStruct)
{
	if (CDialog::OnCreate(lpCreateStruct) == -1)
		return -1;

	return 0;
}




//
// If there is a error, give the right message
//
void CSerialPort::ProcessErrorMessage(char* ErrorText)
{
	char *Temp = new char[200];

	LPVOID lpMsgBuf;

	FormatMessage(
		FORMAT_MESSAGE_ALLOCATE_BUFFER |
FORMAT_MESSAGE_FROM_SYSTEM,
		NULL,
		GetLastError(),
		MAKELANGID(LANG_NEUTRAL, SUBLANG_DEFAULT), // Default language
		(LPTSTR) &lpMsgBuf,
		0,
		NULL
	);

	sprintf(Temp, "WARNING:	%s Failed with the following error: \n%s\nPort: %d\n",
(char*)ErrorText, lpMsgBuf, m_nPortNr);
	MessageBox(NULL, Temp, "Application Error", MB_ICONSTOP);

	LocalFree(lpMsgBuf);
	delete[] Temp;
}

//
```

```
// Write a character.
//
void CSerialPort::WriteChar(CSerialPort* port)
{
    BOOL bWrite = TRUE;
    BOOL bResult = TRUE;

    DWORD BytesSent = 0;

    ResetEvent(port->m_hWriteEvent);

    // Gain ownership of the critical section
    EnterCriticalSection(&port->m_csCommunicationSync);

    if (bWrite)
    {
        // Initailize variables
        port->m_ov.Offset = 0;
        port->m_ov.OffsetHigh = 0;

        // Clear buffer
        PurgeComm(port->m_hComm, PURGE_RXCLEAR | PURGE_TXCLEAR |
PURGE_RXABORT | PURGE_TXABORT);

        bResult = WriteFile(port->m_hComm,
        // Handle to COMM Port
                            port->m_szWriteBuffer,
                // Pointer to message buffer in calling finction
                            port->m_szWriteBufferSize,
/*
    strlen((char*)port->m_szWriteBuffer),*/    // Length of message to send
                            &BytesSent,
                        // Where to store the number of bytes sent
                            &port->m_ov);
                    // Overlapped structure

        // deal with any error codes
        if (!bResult)
        {
            DWORD dwError = GetLastError();
            switch (dwError)
            {
                case ERROR_IO_PENDING:
                    {
                        // continue to GetOverlappedResults()
                        BytesSent = 0;
                        bWrite = FALSE;
                        break;
                    }
                default:
                    {
                        // all other error codes
                        port->ProcessErrorMessage("WriteFile()");
                    }
            }
        }
```

```
                else
                {
                        LeaveCriticalSection(&port->m_csCommunicationSync);
                }
        } // end if(bWrite)

        if (!bWrite)
        {
                bWrite = TRUE;

                bResult = GetOverlappedResult(port->m_hComm,    // Handle to COMM port

&port->m_ov,            // Overlapped structure

&BytesSent,             // Stores number of bytes sent
                                                                          TRUE);
                        // Wait flag

                LeaveCriticalSection(&port->m_csCommunicationSync);

                // deal with the error code
                if (!bResult)
                {
                        port->ProcessErrorMessage("GetOverlappedResults() in WriteFile()");
                }
        } // end if (!bWrite)

        // Verify that the data size send equals what we tried to send
//      if (BytesSent != strlen((char*)port->m_szWriteBuffer))
        if (BytesSent != port->m_szWriteBufferSize)
        {
                TRACE("WARNING: WriteFile() error.. Bytes Sent: %d; Message Length: %d\n",
BytesSent, strlen((char*)port->m_szWriteBuffer));
        }
}

//
// Character received. Inform the owner
//
void CSerialPort::ReceiveChar(CSerialPort* port, COMSTAT comstat)
{
        BOOL    bRead = TRUE;
        BOOL    bResult = TRUE;
        DWORD dwError = 0;
        DWORD BytesRead = 0;
        unsigned char RXBuff;

        for (;;)
        {
                if(WaitForSingleObject(port->m_hShutdownEvent,0)==WAIT_OBJECT_0)
                        return;
                // Gain ownership of the comm port critical section.
                // This process guarantees no other part of this program
                // is using the port object.

                EnterCriticalSection(&port->m_csCommunicationSync);
                // ClearCommError() will update the COMSTAT structure and
                // clear any other errors.

                bResult = ClearCommError(port->m_hComm, &dwError, &comstat);

                LeaveCriticalSection(&port->m_csCommunicationSync);

                // start forever loop.   I use this type of loop because I
                // do not know at runtime how many loops this will have to
                // run. My solution is to start a forever loop and to
                // break out of it when I have processed all of the
                // data available.   Be careful with this approach and
                // be sure your loop will exit.
                // My reasons for this are not as clear in this sample
                // as it is in my production code, but I have found this
                // solutiion to be the most efficient way to do this.

                if (comstat.cbInQue == 0)
                {
                        // break out when all bytes have been read
                        break;
                }

                EnterCriticalSection(&port->m_csCommunicationSync);

                if (bRead)
                {
                        bResult = ReadFile(port->m_hComm,               // Handle to COMM port
                                                           &RXBuff,
                // RX Buffer Pointer
                                                           1,
                // Read one byte
                                                           &BytesRead,
        // Stores number of bytes read
                                                           &port->m_ov);           //
pointer to the m_ov structure
                        // deal with the error code
                        if (!bResult)
                        {
                                switch (dwError = GetLastError())
                                {
                                        case ERROR_IO_PENDING:
                                                {
                                                        // asynchronous i/o is still in
progress
                                                        // Proceed on to
GetOverlappedResults();
                                                        bRead = FALSE;
                                                        break;
                                                }
                                        default:
                                                {
                                                        // Another error has occured.
Process this error.
```

```
    port->ProcessErrorMessage("ReadFile()");
                            break;
                        }
                }
            }
            else
            {
                // ReadFile() returned complete. It is not necessary to call
GetOverlappedResults()
                bRead = TRUE;
            }
        }    // close if (bRead)

        if (!bRead)
        {
            bRead = TRUE;
            bResult = GetOverlappedResult(port->m_hComm,    // Handle to COMM port

&port->m_ov,                // Overlapped structure

&BytesRead,                 // Stores number of bytes read

TRUE);                      // Wait flag

            // deal with the error code
            if (!bResult)
            {
                port->ProcessErrorMessage("GetOverlappedResults() in
ReadFile()");
            }
        }    // close if (!bRead)

        LeaveCriticalSection(&port->m_csCommunicationSync);

        // notify parent that a byte was received
        ::SendMessage((port->m_pOwner)->m_hWnd, WM_COMM_RXCHAR, (WPARAM)
RXBuff, (LPARAM) port->m_nPortNr);

    } // end forever loop

}

//
// Write a string to the port
//
void CSerialPort::WriteToPort(char* string)
{
    assert(m_hComm != 0);

    memset(m_szWriteBuffer, 0, sizeof(m_szWriteBuffer));
    strcpy(m_szWriteBuffer, string);

    // set event for write
    SetEvent(m_hWriteEvent);
}
```

```
void CSerialPort::WriteToPort(char* buf, UINT uSize)
{
        assert(m_hComm != 0);

           memcpy(m_szWriteBuffer, buf, uSize);
           m_szWriteBufferSize=uSize;
        // set event for write
        SetEvent(m_hWriteEvent);
}

//
// Return the device control block
//
DCB CSerialPort::GetDCB()
{
        return m_dcb;
}

//
// Return the communication event masks
//
DWORD CSerialPort::GetCommEvents()
{
        return m_dwCommEvents;
}

//
// Return the output buffer size
//
DWORD CSerialPort::GetWriteBufferSize()
{
        return m_nWriteBufferSize;
}


//-----------------------------------------------------------------------------
void CSerialPort::ClosePort()
{
        // if the thread is alive: Kill
        if (m_bThreadAlive)
        {
                MSG message;
                while (m_bThreadAlive)
                {
                        if(::PeekMessage(&message,m_pOwner->m_hWnd,0,0,PM_REMOVE))
                        {
                                ::TranslateMessage(&message);
                                ::DispatchMessage(&message);
                        }
                        SetEvent(m_hShutdownEvent);
                }
                TRACE("Thread ended\n");

        }
        if(m_szWriteBuffer != NULL)
        {
```

```
        delete [] m_szWriteBuffer;
        m_szWriteBuffer= NULL;
    }

    if(m_hComm)
    {
        CloseHandle(m_hComm);
        m_hComm = NULL;
    }
}


#include "stdafx.h"
#include "MasterGlobalVariables.h"
#include "MasterGlobalFunctions.h"
#include <afx.h>
#include <odbcinst.h>
#include "Serialport.h"
#include "DlgMasterSever.h"

void DataBaseInit(void);
void DataBaseQuit(void);

/*************************************************************************/
/*          save data thread
*/
/*************************************************************************/

UINT SaveDataThread(LPVOID lpParam)
{
    CString strSql;
    UINT Res = 0;
    int tmp;

    unsigned int nRound = 0;

    while (!gbSaveDataThreadBreak)
    {
                    tmp = gTotalCnt;
                    if (tmp == DATANUM)
                    {
                                for(int i=0;i<tmp; i++)
                                {

    strSql.Format(_T("INSERT INTO
M(START,CRC,STPPRD,STPSPD,STPDEG,STPDIR,DCSPD,DCDEG,DCDIR,SWCLAMP,LC,TQ,SEN1,SE
N2,SEN3,STOP) VALUES(%d,%d,%d,%d,%d,%d,%d,%d,%d,%d,%d,%d,%d,%d,%d,%d)")\

    ,(int)MasterStatus[i][0],(int)MasterStatus[i][1],(int)MasterStatus[i][2],(int)MasterStatus[i][3],(int)Mast
erStatus[i][4],(int)MasterStatus[i][5],(int)MasterStatus[i][6],(int)MasterStatus[i][7],(int)MasterStatus[i][8]\

    ,(int)MasterStatus[i][9],(int)MasterStatus[i][10],(int)MasterStatus[i][11],(int)MasterStatus[i][12],(int)
MasterStatus[i][13],(int)MasterStatus[i][14],(int)MasterStatus[i][15]);

    DataBase.ExecuteSQL(strSql);
```

```
		strSql.Format(_T("INSERT INTO
S(START,CRC,STPPRD,STPSPD,STPDEG,STPDIR,DCSPD,DCDEG,DCDIR,SWCLAMP,LC,TQ,SEN1,SEN
2,SEN3,STOP) VALUES(%d,%d,%d,%d,%d,%d,%d,%d,%d,%d,%d,%d,%d,%d,%d,%d)")\

		,(int)SlaveStatus[i][0],(int)SlaveStatus[i][1],(int)SlaveStatus[i][2],(int)SlaveStatus[i][3],(int)SlaveStat
us[i][4],(int)SlaveStatus[i][5],(int)SlaveStatus[i][6],(int)SlaveStatus[i][7],(int)SlaveStatus[i][8]\

		,(int)SlaveStatus[i][9],(int)SlaveStatus[i][10],(int)SlaveStatus[i][11],(int)SlaveStatus[i][12],(int)Slave
Status[i][13],(int)SlaveStatus[i][14],(int)SlaveStatus[i][15]);

		DataBase.ExecuteSQL(strSql);
									}

		//////////////////////////////////////////////////////////////////////
								//		gTotalCnt = 0;

		AfxMessageBox(_T("Finished."), MB_OK, NULL);
					}
		}

		DataBaseQuit();
		Res = 1;
		return Res;
}

void DataBaseInit(void)
{
		CString strSql;
		CTime CurTime = CTime::GetCurrentTime();
		CString strTime = CurTime.Format("%m-%d-%H-%M-%S");
		CString strTmp;
		strTmp.Format(_T(" F-%d"),nExcelFileCnt);
		CString strExcelFileName =_T( "C:\\Datafile\\") + strTime+strTmp + _T(".xls");
		CString strDriver("MICROSOFT EXCEL DRIVER (*.XLS)");

		TRY
		{

		strSql.Format(_T("DRIVER={%s};DSN="";FIRSTROWHASNAMES=1;READONLY=FALSE;CR
EATE_DB=\"%s\";DBQ=%s"),
					strDriver, strExcelFileName, strExcelFileName);

				if (DataBase.OpenEx(strSql,CDatabase::noOdbcDialog))
				{
					strSql= "CREATE TABLE M(START NUMBER,CRC NUMBER,STPPRD
NUMBER,STPSPD NUMBER,STPDEG NUMBER,STPDIR NUMBER,DCSPD NUMBER,DCDEG
NUMBER,DCDIR NUMBER,SWCLAMP NUMBER,LC NUMBER,TQ NUMBER,SEN1 NUMBER,SEN2
NUMBER,SEN3 NUMBER,STOP NUMBER)";
					DataBase.ExecuteSQL(strSql);
					strSql= "CREATE TABLE S(START NUMBER,CRC NUMBER,STPPRD
NUMBER,STPSPD NUMBER,STPDEG NUMBER,STPDIR NUMBER,DCSPD NUMBER,DCDEG
NUMBER,DCDIR NUMBER,SWCLAMP NUMBER,LC NUMBER,TQ NUMBER,SEN1 NUMBER,SEN2
NUMBER,SEN3 NUMBER,STOP NUMBER)";
					DataBase.ExecuteSQL(strSql);
				}
```

```
        }
        CATCH_ALL (e)
        {
                        TRACE("Excel driver is not installed: %s", strDriver);
        }
        END_CATCH_ALL

        nExcelFileCnt++;
}

void DataBaseQuit(void)
{
        if (DataBase.IsOpen())
        {
                DataBase.Close();
        }
}



UINT SerialPortTXDataThread(LPVOID lParam)
{
        HWND hwnd = (HWND)lParam;

        while(1)
        {
                WaitForSingleObject(g_COM_Event, INFINITE);
                ::SendMessage(hwnd, WM_MSG_COM_TX, NULL, NULL);
                g_COM_Event.ResetEvent();
//              Sleep(100);
        }

        return 0;
}

UINT LanTXDataThread(LPVOID lParam)
{
        HWND hwnd = (HWND)lParam;

        while(1)
        {
                WaitForSingleObject(g_LAN_Event, INFINITE);            //        ?
                ::SendMessage(hwnd, WM_MSG_LAN_TX, NULL, NULL);
                g_LAN_Event.ResetEvent();
//              Sleep(100);
        }
        return 0;
}

//------------------------------------------------------------------------------------------------------------------
//
//
                Global variables used in MasterData
//
                        [2/22/2011 Administrator]
```

```
//
                                             MasterGlobalVariables.cpp
//-------------------------------------------------------------------------------------------------------------------
#include "stdafx.h"
#include <afxdb.h>
#include "MasterGlobalVariables.h"
#include "CMSComm.h"

LONG gGpCnt;
LONG gTotalCnt;

// BYTE ByteBuffer[128];
unsigned int MasterStatus[DATANUM][17];
unsigned int SlaveStatus[DATANUM][17];

BYTE MasterSTFromCom[16];
int g_MSTFromCom;
char SlaveSTFromLan[16];
int g_SSTFromCom;

//--------------------------------------------------------------------------------------------------------------------
unsigned char gIsDrawCircle;
unsigned char gIsDrawStart;

//--------------------------------------------------------------------------------------------------------------------
CDatabase DataBase;
unsigned int nExcelFileCnt;

CWinThread* ThreadSaveData;
// CWinThread* ThreadSerialPort;

BOOL gbSaveDataThreadBreak;
BOOL gbSerialPortThreadBreak;

unsigned char gIsSaveFinished;

char recvbuf[SOCK_BUF_LEN];
char sendbuf[SOCK_BUF_LEN];


//--------------------------------------------------------------------------------------------------------------------
//                              serial port

CEvent g_COM_Event;
CEvent g_LAN_Event;



// MasterDataDoc.cpp : CMasterDataDoc  ク¤¡¥ラ¤¨ス¤1の è?実g装Á ¡ã
//

#include "stdafx.h"
#include "MasterData.h"

#include "MasterDataDoc.h"
```

```
#ifdef _DEBUG
#define new DEBUG_NEW
#endif


// CMasterDataDoc

IMPLEMENT_DYNCREATE(CMasterDataDoc, CDocument)

BEGIN_MESSAGE_MAP(CMasterDataDoc, CDocument)
END_MESSAGE_MAP()



CMasterDataDoc::CMasterDataDoc()
{

}

CMasterDataDoc::~CMasterDataDoc()
{
}

BOOL CMasterDataDoc::OnNewDocument()
{
	if (!CDocument::OnNewDocument())
		return FALSE;

	return TRUE;
}
// CMasterDataDoc シ¤¡¤   ¤?E v  é£¤꾀E?

void CMasterDataDoc::Serialize(CArchive& ar)
{
	if (ar.IsStoring())
	{

	}
	else
	{

	}
}


#ifdef _DEBUG
void CMasterDataDoc::AssertValid() const
{
	CDocument::AssertValid();
}

void CMasterDataDoc::Dump(CDumpContext& dc) const
{
	CDocument::Dump(dc);
}
#endif //_DEBUG
```

```
#include "stdafx.h"
#include "MasterData.h"
#include "MainFrm.h"

#include "MasterDataDoc.h"
#include "MasterDataView.h"
#include "CMSComm.h"
#include "MasterGlobalFunctions.h"
#include "DlgDataReceive.h"
#include "DlgConnectServer.h"

#ifdef _DEBUG
#define new DEBUG_NEW
#endif


// CMasterDataApp

BEGIN_MESSAGE_MAP(CMasterDataApp, CWinApp)
    ON_COMMAND(ID_APP_ABOUT, &CMasterDataApp::OnAppAbout)
    ON_COMMAND(ID_FILE_NEW, &CWinApp::OnFileNew)
    ON_COMMAND(ID_FILE_OPEN, &CWinApp::OnFileOpen)
    ON_COMMAND(ID_FILE_PRINT_SETUP, &CWinApp::OnFilePrintSetup)
END_MESSAGE_MAP()


CMasterDataApp::CMasterDataApp()
{

}

CMasterDataApp theApp;


BOOL CMasterDataApp::InitInstance()
{
    INITCOMMONCONTROLSEX InitCtrls;
    InitCtrls.dwSize = sizeof(InitCtrls);
    InitCtrls.dwICC = ICC_WIN95_CLASSES;
    InitCommonControlsEx(&InitCtrls);

    CWinApp::InitInstance();

    if (!AfxOleInit())
    {
        AfxMessageBox(IDP_OLE_INIT_FAILED);
        return FALSE;
    }
    AfxEnableControlContainer();
    LoadStdProfileSettings(4);
    CSingleDocTemplate* pDocTemplate;
    pDocTemplate = new CSingleDocTemplate(
        IDR_MAINFRAME,
        RUNTIME_CLASS(CMasterDataDoc),
        RUNTIME_CLASS(CMainFrame),       // メ ¤¨¢イ ¤¡è   ¤?ESDI  フ¤?   ¤?E`   ¤?Eウ
¤イ ¤¡êン ¤¨® ド ¤¨¦ウ ¤
        RUNTIME_CLASS(CMasterDataView));
    if (!pDocTemplate)
        return FALSE;
    AddDocTemplate(pDocTemplate);

    CCommandLineInfo cmdInfo;
    ParseCommandLine(cmdInfo);

    if (!ProcessShellCommand(cmdInfo))
        return FALSE;

    m_pMainWnd->ShowWindow(SW_SHOW);
    m_pMainWnd->UpdateWindow();
    return TRUE;
}


class CAboutDlg : public CDialog
{
public:
    CAboutDlg();

    enum { IDD = IDD_ABOUTBOX };

protected:
    virtual void DoDataExchange(CDataExchange* pDX);    protected:
    DECLARE_MESSAGE_MAP()
};

CAboutDlg::CAboutDlg() : CDialog(CAboutDlg::IDD)
{
}

void CAboutDlg::DoDataExchange(CDataExchange* pDX)
{
    CDialog::DoDataExchange(pDX);
}

BEGIN_MESSAGE_MAP(CAboutDlg, CDialog)
END_MESSAGE_MAP()

void CMasterDataApp::OnAppAbout()
{
    CAboutDlg aboutDlg;
    aboutDlg.DoModal();
}

#include "stdafx.h"
#include "ListenSocket.h"
#include "DlgMasterSever.h"

CListenSocket::CListenSocket(): m_bIsConnected(FALSE)
```

```
{

}

CListenSocket::~CListenSocket()
{

}

void CListenSocket::OnAccept(int nErrorCode)
{
        CServerSocket *psk = new CServerSocket();
        if (Accept(*psk))
        {
                                        //      AfxMessageBox(_T("Connecting request"));
                                                m_SeverSocket = psk;
                                                m_SeverSocket->m_hwnd = m_hwnd;
                                                m_bIsConnected = true;
                                                ::SendMessage(m_hwnd,
WM_MSG_LAN_RX, SOCK_CLIENT_REQUEST, NULL);
//                                              PostMessage(m_hwnd, WM_MYMSG,
SOCK_CLIENT_REQUEST, NULL);
                                                CSocket::OnAccept(nErrorCode);
                        }else
                        {
                                                delete      psk;
                        }
                        AsyncSelect( FD_READ|FD_WRITE|FD_CLOSE );
}
```

Appendix II

The program code of the robot-assisted catheter system (CCStudio v3.3)

The program code of the master system:

```
# include "DSP2833x_Device.h"
#define MASTER_SIDE 1
#define SLAVE_SIDE 0
//----------------------------------------------------------------------------
volatile Uint16 Flag_1ms; // Define the main loop timer flag
volatile Uint16 GlobalDelayCount;        //        Global delay
//----------------------------------------------------------------------------
//        //        variables from GPIO
//        Uint16 SteppingTim1;    //        GPIO76
Uint16 SteppingAlarm;   //        GPIO75
Uint16 SteppingEnd;     //        GPIO74
Uint16 ExSw1;   //        GPIO68
Uint16 ExSw10; //        Last status of sw1
Uint16 ExSw2;   //        GPIO67
Uint16 ExSw20; // Last status of Sw2
Uint16 ExSw3;   //        GPIO66
Uint16 ExSw4;   //        GPIO65
Uint16 ExSw5;   //        GPIO64
```

```
Uint16 LimitSw1;              //        GPIO46

Uint16 LimitSw2;              //        GPIO47

Uint16 LimitSw3;              //        GPIO80

//-------------------------------------------------------------------

//    sensor variables from adc

//

// float LoadCell;//            ADCA3

// Uint16 LoadCellOffset;

/*

float Torque;        //         ADCA1

float Pressure0;     //         ADCA2

float Pressure1;     //         ADCA3

float Pressure2;     //         ADCA4

float VariableRegister0;      //        ADCA5

float VariableRegister1;      //        ADCA6

float Reserve;       //         ADCA7

*/

//-------------------------------------------------------------------

//          // output variables for GPIO

//

Uint16 SteppingACL;           //        GPIO79

Uint16 Steppingx10;           //        GPIO78
```

```
Uint16 SteppingCOFF;    //          GPIO77

Uint16 MaxonDacCs;      //          GPIO73

Uint16 MaxonDacSck;     //          GPIO72

Uint16 MaxonDacSdi;     //          GPIO71

Uint16 MaxonDacLdac;    //          GPIO70

Uint16 MaxonEN;         //          GPIO69

Uint16 LED1;    //          GPIO40

Uint16 LED2;    //          GPIO41

Uint16 LED3;    //          GPIO42

Uint16 LED4;    //          GPIO43

//---------------------------------------------------------------------

//          // output variables for EQEP

//POSSPEED qepSTP_posspeed = POSSPEED_DEFAULTS;

POSSPEED qepDC_posspeed = POSSPEED_DEFAULTS;

//---------------------------------------------------------------------

// Global DATA

// Spi communication data

//

Uint16 MasterStatus[DATALEN];

Uint16 SlaveStatus[DATALEN];

Uint16 SCI_CNT;

Uint16 SCI_COPY;
```

```
char SciCopyBuff[2];

Uint16 gTest;

volatile Uint16 BusyFlag;

//--------------------------------------------------------------------

// Control variable

//

float StpV0;

int DcDeg0;               //        Q15

int StpDeg0;      //        Q15

#include "DSP2833x_Device.h"

#define MASTER_SIDE 1

#define SLAVE_SIDE 0

//--------------------------------------------------------------------

//

//                                          *         Device Initial    *

//                                                    (Both M & S)

// //--------------------------------------------------------------------

////                        Stepping initial

//void InitStepping(void)

{

        Uint16 i;

        GpioDataRegs.GPCSET.bit.GPIO79= 1;   //        Clear Alarm      Gpio79= 1;
        //DELAY_US(1L);
        for(i=0;i<500;i++){}
        GpioDataRegs.GPCCLEAR.bit.GPIO79= 1;        // Set alarm to low; Gpio79=
0
                // GpioDataRegs.GPCCLEAR.bit.GPIO78= 1;     // Set x10; 1 - x10;
0 - none;       Gpio78= 0;
        // GpioDataRegs.GPCCLEAR.bit.GPIO77= 1;     // Stepping C.Off; 1 - off; 0 -
on;     Gpio77= 0;
}
//---------------------------------------------
//                          DAC Initial
void InitDAC(void)
{
        //      Other pins was set in Mcbspb initial
        GpioDataRegs.GPCCLEAR.bit.GPIO70= 1;        //      Set #LDAC to low,
P2-12
}
/---------------------------------------------
//              Global variables initial
void InitGlobalVars(void)
{
        Uint16 i;
```

```
// Status initial
	for(i=0;i<DATALEN;i++)
	{
		MasterStatus[i] = 0x0000;
		SlaveStatus[i] = 0x0000;
	}
//	MasterStatus[STR] = 0x0808;
	MasterStatus[STR] = 0xABCD;
	MasterStatus[CRC]= 0x0888;
//	MasterStatus[STOP]= 0x0C0C;
	MasterStatus[STOP] = 0xDCBA;
//	Ex switch for clamp
	ExSw1 = 0;
	ExSw10 = 0;
	ExSw2 = 0;
	ExSw20 = 0;
// Qep posspeed var initial
	qepSTP_posspeed.DevSel = STEPPING;
	qepDC_posspeed.DevSel = DC;

	qepSTP_posspeed.mech_scaler= 16776;		//	Q26
	qepDC_posspeed.mech_scaler= 512;			//		Q0
	qepSTP_posspeed.theta_raw= 0;
	qepDC_posspeed.theta_raw= 0;
	qepSTP_posspeed.init(&qepSTP_posspeed);
	qepDC_posspeed.init(&qepDC_posspeed);
// Global Delay Count
	GlobalDelayCount = 0;
	SCI_CNT = 8;
	SCI_COPY = 0;
	gTest = 0;
	StpV0= 0.0;
	StpDeg0 = 0;	// Q15
	DcDeg0= 0;		// Q15
}
//---------------------------------------------------------------------
////	******************	Functions used in Master side	******************
// //---------------------------------------------------------------------
#if MASTER_SIDE
//----------------------------------
//				Step 1
void GetUserStatus_M(void)
{
        GetADC();       //      User's input was got
        GetGpio();      //      Control panel, Limit switch
        GetEQep();      //      Stepping displacement and speed, DC rolling angle
and speed

//      DataSendToSlave[STPSPD] = qepSTP_posspeed.Speed_fr
}
//-----------------------------------
//                      Step 2
void SendCMDtoSlave_M(void)
{
/*---------------------------------------------------
        Uint16 i;
        Uint16 tmp;
        if(GpioDataRegs.GPBDAT.bit.GPIO58 == 0)
        {
                for(i=0;i<DATALEN;i++)
                {
                        while(SpiaRegs.SPISTS.bit.BUFFULL_FLAG == 1){}
                        SpiaRegs.SPITXBUF= MasterStatus[i];
                        while(SpiaRegs.SPISTS.bit.INT_FLAG == 0) {}
                        tmp = SpiaRegs.SPIRXBUF;
```

```
            if(i != 0)
            {
                SlaveStatus[i-1] = tmp;
            }
        }

    }
------------------------------------------------------*/
}
//-----------------------------------
//      Step 3
void SetMasterStatus_M(void)
{

//  float tmp;
//-------------------------
//  Set Stepping
    if(GlobalDelayCount % 10 == 0)
    {
            SetMasterStepping();
        }
//-------------------------
```

```
//        Set Maxon through DAC
//        tmp = 3.0*SlaveStatus[LC]/4096.0;
          SetDAC(1.65,0); //        0-A, 1-B
          // just for test
}
#endif    //        END of MASTER_SIDE
//-----------------------------------------------------------------------
//
//                                      *         Sub functions     *
//                                                (Both M & S)
//-----------------------------------------------------------------------
//
//        Get user information from ADC
//
void GetADC(void)
  {

          AdcRegs.ADCTRL2.bit.RST_SEQ1= 1;              // Reset SEQ1
          // AdcRegs.ADCTRL2.bit.RST_SEQ2= 1;           // Reset SEQ2
          //        Start SEQ1
    AdcRegs.ADCTRL2.all = 0x2000;
          while (AdcRegs.ADCST.bit.INT_SEQ1== 0) {} // Wait for interrupt
          AdcRegs.ADCST.bit.INT_SEQ1_CLR= 1; // Clear SEQ1 IF

          // AdcRegs.ADCST.bit.INT_SEQ2_CLR= 1;


          // Get sensors status from ADC

                    MasterStatus[LC]= (AdcRegs.ADCRESULT3>>4);

          //        (float)(3.0*((AdcRegs.ADCRESULT3>>4)/4095.0)) - LoadCellOffset ;

          // ADC0 Get Load Cell;

/*                  Torque= (AdcRegs.ADCRESULT1>>4);    // ADC1 Torque

                    Pressure0= (AdcRegs.ADCRESULT2>>4);                    // ADC2

Pressure0

                    Pressure1= (AdcRegs.ADCRESULT3>>4);                    // ADC3

Pressure1

                    Pressure2= (AdcRegs.ADCRESULT4>>4);                    // ADC4

Pressure2

                    VariableRegister0= (AdcRegs.ADCRESULT5>>4);            // ADC5

VR0

                    VariableRegister1= (AdcRegs.ADCRESULT6>>4);            // ADC6

VR0

*/


  }
```

```
//-----------------------------------------------------
//          Get user information from GPIO
void GetGpio(void)
{
          SteppingTim1= GpioDataRegs.GPCDAT.bit.GPIO76;
          SteppingAlarm= GpioDataRegs.GPCDAT.bit.GPIO75;
          SteppingEnd= GpioDataRegs.GPCDAT.bit.GPIO74;

//          ExSw1= GpioDataRegs.GPCDAT.bit.GPIO68;
//          ExSw2= GpioDataRegs.GPCDAT.bit.GPIO67;
          ExSw3= GpioDataRegs.GPCDAT.bit.GPIO66;
          ExSw4= GpioDataRegs.GPCDAT.bit.GPIO65;
          ExSw5= GpioDataRegs.GPCDAT.bit.GPIO64;

          LimitSw1= GpioDataRegs.GPBDAT.bit.GPIO46;
          LimitSw2= GpioDataRegs.GPBDAT.bit.GPIO47;
          LimitSw3= GpioDataRegs.GPCDAT.bit.GPIO80;

// --------Get Clamp Sw State -------------
          if( GlobalDelayCount % 100 == 0)
          {
                if((GpioDataRegs.GPCDAT.bit.GPIO68 == 0) && (ExSw10 == 0))
                {
                    ExSw1 = (~ExSw1) & (0x0001);              // change last bit
                    ExSw10 = 1;
                }
                if(GpioDataRegs.GPCDAT.bit.GPIO68 == 1)
                {
                        ExSw10 = 0;
                }
                if((GpioDataRegs.GPCDAT.bit.GPIO67 == 0) && (ExSw20 == 0))
                {
                    ExSw2 = (~ExSw2) & (0x0001);              // change last bit
                    ExSw20 = 1;
                }
                if(GpioDataRegs.GPCDAT.bit.GPIO67 == 1)
                {
                    ExSw20 = 0;
                }
        }
                MasterStatus[SWCLAMP] = ExSw1;
//-----------------------------------------
        /*---------------------*/
```

```
//          BusyFlag = GpioDataRegs.GPCDAT.bit.GPIO83;              // SPI BUSY
//          BusyFlag = 0;    //          just for test, comment it when
finished//////////////////////////////////////
}
//------------------------------------------------------------
// Get EQep information position & speed of stepping & DC
void GetEQep(void)
{
            int tmp;
            qepSTP_posspeed.calc(&qepSTP_posspeed);
            qepDC_posspeed.calc(&qepDC_posspeed);
            ///////////
//          EQep1Regs.QEPCTL.bit.SWI = 1;
            MasterStatus[STPDEG] = qepSTP_posspeed.theta_raw;
            MasterStatus[STPSPD] = qepSTP_posspeed.Speed_pr;                            //
_iq, Q0
            MasterStatus[STPDIR] = qepSTP_posspeed.DirectionQep;
            ///////////
            tmp = DcDeg0 + qepDC_posspeed.theta_raw;
            DcDeg0 = tmp;
            EQep2Regs.QEPCTL.bit.SWI = 1;
            MasterStatus[DCDEG] = tmp;
```

```
        MasterStatus[DCSPD] = qepDC_posspeed.Speed_pr;          //_q, Q0
        MasterStatus[DCDIR]     = qepDC_posspeed.DirectionQep;
        /*
        STPDisp= qepSTP_posspeed.theta_mech * STPPITCH;     // Q15 * Q0 = Q15,
displacement= angle*pitch
        STPSpeed= qepSTP_posspeed.Speed_pr;                   // _iq, Q0
        STPDir= qepSTP_posspeed.DirectionQep;
        DCDisp= qepDC_posspeed.theta_mech;            // Q15
        DCSpeed= qepDC_posspeed.Speed_pr;             // _iq, Q0
        DCDir= qepDC_posspeed.DirectionQep;
*/
}
//---------------------------------------------------------
//      SetDAC voltage
//      DAC is MCP4922, 16 bits command word, bits11:0 are data bits
//      bit15: 0: DAC_B, 1: DAC_A
//      bit14: 0: Unbuffered Vref, 1: buffered Vref
//      bit13: 0: (Vout=2*Vref*D/4096), 1: (Vout=1*Vref*d/4096)
//      bit12: #SHDN, 0: output buffer disable, high impedance, 1: Output power donw
control bit
//---------------------------------------------------------
void SetDAC(float voltage, int Sel)
```

```
{
        Uint16 Tmp;
        if(Sel == 0)            //      DAC-A bit15 = 0
        {
                Tmp= (Uint16)(4096*voltage/3.35) & 0x0FFF | 0x3000;
        }else if(Sel == 1)
        {
                Tmp= (Uint16)(4096*voltage/3.35) & 0x0FFF | 0xA000;
        }
        McbspbRegs.DXR1.all= Tmp;
}
//---------------------------------------------------------
//      Set Master Stepping
//      Direction, Frequency
//
//---------------------------------------------------------
void SetMasterStepping(void)
{
        float dF,v, Fl, C0, tmp;
        float Fs;
        Uint16 tbprd;
        C0 = 0.001;
        //      Just for test comment when release
        if(SlaveStatus[STR] == 0xABCD)
        {
                Fs = (SlaveStatus[LC]*3.0/4096.0)-SLC_VOLTAGE_OFS;
        }
        Fl= (((int)MasterStatus[LC])*3.0/4096) - MLC_VOLTAGE_OFS;
//      Fl= (((int)MasterStatus[LC])*3.0/4096);
        if((Fl<0.1) && (Fl>(-0.1)))
        {
                dF = 0;
        }else{
//              dF= Fl + 0.5* Fs;
                dF = Fl;
                dF*= 5/LC_VOLTAGE_RANGE;
        }
        tmp = 1000* MasterStatus[STPSPD]/1500000.0;
        C0 = C0 + tmp;
        v= 5*MASTER_M * C0*dF*1.55 + 0.5*StpV0;
        StpV0= v;
        if((Fl>0.1)||(Fl<(-0.1)))
        {
                if(dF>0)
        {

            if(Fl>0)
            {
                tbprd = (int)(1500/v - 1);
                if(tbprd<1000)    // MAX frq is 15K
                {
                    tbprd = 1000;
                }
                EPwm1Regs.TBPRD= tbprd;
                EPwm1Regs.CMPA.half.CMPA= (tbprd)/2;            //
Duty cycle= 50%
                EPwm1Regs.CMPB = (tbprd)/2;    // Duty cycle= 50%

                STOPEPWM2();
                STARTEPWM1();
                MasterStatus[STPPRD] = tbprd;
                MasterStatus[STPDIR] =0;
            //  MasterStatus[STPDIR] = 1;
            }else if(Fl<0)
            {
                MasterStatus[STPPRD] = 0;
                STOPEPWM1();
                STOPEPWM2();
            }
        }else if(dF<0)
        {
            if(Fl<0)
            {
                tbprd = (-1)*((int)1500/v-1);
                if(tbprd<1000)
                {
                    tbprd = 1000;
                }
                EPwm2Regs.TBPRD= tbprd;
                EPwm2Regs.CMPA.half.CMPA= (tbprd)/2;        // Duty cycle= 50%
                EPwm2Regs.CMPB = (tbprd)/2;    // Duty cycle= 50%
                STOPEPWM1();
                STARTEPWM2();
                MasterStatus[STPPRD] = tbprd;
                MasterStatus[STPDIR] = 1;
            }else if(Fl>0)
            {
                MasterStatus[STPPRD] = 0;

                STOPEPWM1();

                STOPEPWM2();

            }

        }

    }else

    {

        MasterStatus[STPPRD] = 0;

        STOPEPWM1();

        STOPEPWM2();

    }
```

Biographic Sketch

Jin Guo received his B.Sc. degree and his M.Sc. degree from the University of Electronic Science and Technology of China in 2008 and 2011, China. Currently, he is a Ph.D. candidate in Kagawa University, Japan. He researches on robotic catheter systems, virtual reality technology, medical image processing, and haptic devices.

He has published about 9 refereed journals and conference papers in recent three years.