Imperial College London
Department of Computing

Simulation of Blood Flow and Contrast Medium Propagation for a Vascular Interventional Radiology Simulator

By
Yingtao Wang

Submitted in partial fulfilment of the requirements for the MSc Degree in Computing of Imperial College London

September 2009

Abstract

Minimally invasive technique provides a revolutionized clinical therapy, which significantly reduces operation trauma, recovery time, and overall clinical costs. Interventional radiologists use this technique to access vascular systems, and use catheterization to navigate to the region of interest with the help of the medical imaging technique. Medical education and training offer a virtual clinical environment for doctors to have highly qualified professional skills, and to provide high-quality care to patients.

This project is mainly based on a framework of virtual catheterisation simulator (VCSim) developed in St Mary's hospital, Imperial College London. This simulator models instruments using a mass-spring model, and provides interfaces with interventional radiology specific haptic devices. Clinicians are able to use this simulator to practice catheterization through different realistic vascular models. However, during virtual fluoroscopy, the vascular surfaces cannot be seen clearly without medical contrast medium which highlights the vessels. With the effect of beating blood flow produced by heart rate, contrast medium mixes and propagates through vasculature and allows visualisation of the blood vessels. This is called an angiography.

This project aims to simulate blood flow and contrast medium propagation in three dimensional virtual vasculatures reconstructed from real patient CT scans. The useful information such as bifurcation and cross-section of vasculature are obtained from the corresponding centreline generated from patient datasets and processed through a centreline reconstruction tool. The blood flow is controlled by a beating heart model, interacting with the contrast medium propagation. The contrast medium is modelled using smoothed-particle hydrodynamics, and is constrained by three forces produced by initial injection, collision with vessel walls and beating blood flow. Moreover, an infinite number of injections are possible, and an initial review system, based on snap shot of the injection, is proposed. Furthermore, this project uses a haptic device to track a real catheter and control its virtual counterpart. On the same model, a real syringe is used to inject virtual contrast medium through the catheter as a clinician would do in real life. The simulation is tested in three different vasculatures, and theoretically supports complex vasculatures with a large number of branches and sub-branches. Further, the simulation is evaluated both by clinicians and through comparison with real injection videos. The result is convincing and can be used as the foundation of a more realistic contrast medium under specific and complex blood flow cases.

This MSc individual project is developed under the supervision of Dr. Fernando Bello and Dr. Vincent Luboz at St Mary's hospital, Imperial College London.

Key Words: Interventional radiology · Vascular modelling · Medical training · Simulation · Blood flow · Contrast medium · Centreline · Haptic device · Smoothed-particle hydrodynamics

Acknowledgements

During the creation of this individual project, I was fortunate to have the great help of many people.

I would like to thank my supervisor Dr. Fernando Bello for giving me the precious chance to be involved in the CRaIVE project, letting me work in the famous St Mary's hospital, and giving me the opportunities to share my ideas in the group meetings. I am grateful for his important advices, comments and suggestions both in the implementation and dissertation writing.

I am much indebted to Dr. Vincent Luboz for his great help on the project. Through hundreds of E-mails, day-to-day meeting, and some tutorials, I have highly benefited from his supervision throughout the whole project. I would also like to thank him for bringing me to Wales to look for a solution of the project, and always spending a lot of precious time on reading and modifying my writing. A lot of his comments, suggestions and invaluable ideas greatly affect my implementation and dissertation writing.

I would also like to thank Dr. Derek Gould who provides professional evaluation for the blood flow and contrast medium propagation simulation, and gives important and detailed feedback on the project. Besides, I learnt a lot of medical knowledge from him which helped improving the fluid modeling and simulation.

I would like to thank Dr. Eddie Edwards and Dr. Pierre Villard for suggestions during fortnight progress meetings, and selfless help during project implementation.

The project would not be accomplished without the help from my friend Mr. Xiaoyuan Zhou. We shared ideas and codes from time to time, and many new ideas are inspired from the frequent discussions.

Finally, I would like to thank my family who gives me great support during the project, and all my friends, classmates and colleagues who helped the development of this project.

Contents

1. Introduction

This chapter provides a broad overview of the project. It poses the motivation of this project, and introduces the basic methods and chosen solution. Further, brief summaries of following chapters of the thesis are given.

1.1 Project overview

My individual project is a part of the real-time vascular interventional radiology simulator of the CRaIVE (Collaboration in Radiological Interventional Virtual Environments) consortium. This simulator is funded by EPSRC. Since 2006, groups of researchers from University of Leeds, University of Bangor, University of Liverpool, University of Hull, Imperial College, Liverpool Hospital, Leeds Hospital and Manchester Hospital are focusing on developing and evaluating different parts of this simulator. These areas include CT scan image processing, vascular segmentation and 3D reconstruction, as well as haptic visual reality platform establishment, real-time visual catheterization simulation, and professional evaluation.

The aim of this simulator is to provide a virtual environment to clinicians for training Seldinger technique skills on many different virtual anatomies from real patients. The Seldinger technique consists in performing a small skin nick to access the vasculature (usually around the femoral artery) and in inserting a needle through the opening followed by a guidewire and a catheter in order to perform an angiography and eventually treating pathology. Key virtual manipulations such as needle puncture, guidewire and catheter navigation need to be modelled. Human interactions in minimal invasive radiological procedures can be accurately felt through the haptic device.

1.2 Problem posing

As an important part of the vascular interventional radiology simulator, the current prototype of a virtual catheterization environment [1] has been developed in St Mary's hospital, Imperial College London. The framework provides realistic interactions between guidewire, catheter and needle, uses complex vascular models segmented from 28 patient datasets, achieves real-time simulation, and combines an interventional radiology specific haptic device. An extension feature of the framework is to simulate medical navigation steps and perform a diagnosis progress under the view of simulated fluoroscopy environment. In order to see the vasculature contours using angiography in the real medical imaging, contrast medium, a viscous fluid usually composed of Iodine, is injected into the blood vascular system through an inserted catheter.

My project aims to simulate real-time blood flow and contrast medium propagation on the foundation of the existing framework to enhance the virtual environment by allowing the trainee to inject contrast medium in the vessels through a modelled virtual catheter.

1.3 Methods and chosen Solution

A successful medical simulation combines knowledge from medicine, bioengineering and computer science. Interventional Radiologists in our group are responsible for setting the task analysis and the metrics, evaluating the simulation and helping reaching a realistic rendering. In order to model a realistic blood flow and contrast agent, existing methods of fluid modelling already published in many bioengineering papers; formulas of classic fluid mechanics could be used in our simulator. Some of these models are selected for the specific project to achieve realistic performance and real-time speed.

My work is mainly in the field of computer programming and implementation. Computer graphics is used to generate a three dimensional vasculature and other geometric objects, computer visualization technique processes and transforms complex biomedical datasets into a virtual expression. Moreover, physics-based rendering technique using fluid models is necessary for a realistic simulation, and smoothed-particle hydrodynamics rendering is used for viscous fluid rendering and propagation.

This project is developed in Microsoft Visual Studio 2005 environment and uses the C++ programming language. Graphics rendering library VTK (http://www.vtk.org/) is used for medical data processing and rendering, light user-interface library FLTK (http://www.fltk.org/) allows building a simple and clear user interface, and a commercial interface connects the framework with the haptic device.

1.4 Composition of the thesis

The thesis includes nine chapters followed by bibliography, appendix and user's guide. The following eight chapters are organized as follow with brief overviews.

Chapter 2: Background

This chapter discusses the basic background of the project in both the clinical view and technical view. The clinical background describes concepts of minimally invasive surgery and interventional radiology, gives basic principles of blood flow and contrast medium propagation which need to be modelled and simulated, and introduces creative notion of medical education and training. The technical background focuses on simulation techniques including graphics, image processing and visualization, physics-based rendering and haptic device, as well as state-of-the-art fluid simulation models. A project methodology using smoothed-particle hydrodynamics rendering is also introduced.

Chapter3: Concept design

This chapter describes the concept design of the simulator, and provides detailed design for the centreline (i.e. the line connecting the cross section centres of the vessels) reconstruction tool, blood flow module and contrast medium module. Chosen technical toolkits are also explained.

Chapter4: Centreline reconstruction tool

The vascular centreline provides prior information for contrast medium propagation. A developed centreline reconstruction tool is described in this chapter. It provides two standard files to the fluid simulation framework. The centreline file includes vascular branches, and skeleton points with radius, position and maximum velocity, while the centreline configuration file information includes bifurcation feature points, bifurcation branches, and control points.

Chapter 5: Blood flow simulation

This chapter provides details on the chosen model for beating heart. Two key factors on contrast medium propagation are also extracted as forms of linear opacity function and blood flow speed changing function.

Chapter 6: Contrast medium simulation

An innovative full set of models of contrast medium is described in this chapter, including propagation modelling, bifurcation and collision detection and response, as well as specific systems of radius-changing, fading, and real-time speed control. The information presented in Chapter 4 and Chapter 5 is combined and used in this set of models and systems.

Chapter 7: User-Interaction

Users can inject contrast agent multiple times in the catheterization training process. Furthermore, the contrast medium can be injected through the VSP interface or with the keyboard and the user interface. A control panel is designed for parameter control over the contrast medium. Different patient cases have been generated to widen the scope of the training, and a simulation review system (based on 3D snapshots) can be used for evaluation and tracking for the whole process of the contrast medium injection.

Chapter 8: Evaluation

This chapter presents an evaluation of the contrast medium simulation as well as the performance of the centreline reconstruction tool, and gives an example with different patient-specific vascular cases. Further, feedback from clinicians and comparison to real injection videos is given for evaluation of the simulation.

Chapter 9: Conclusion and future work

This chapter concludes the whole project, summarising all achievements, discussing its results, and giving recommendations for future work. Mathematical background and abbreviations are outlined in Appendix A and B, respectively. In the end of thesis, a user’s guide is included for guiding the use of the reconstruction tool and the injection application.

2. Background

This chapter discusses the basic background of the project in both the clinical view and technical view. The clinical background describes concepts of minimally invasive surgery and interventional radiology, gives basic principle of blood flow and contrast medium propagation which need to be modelled and simulated, and introduces creative notion of medical education and training. The technical background focuses on simulation techniques including graphics, image processing and visualization, physics-based rendering and haptic device, as well as state-of-the-art fluid simulation models. A project methodology using smoothed-particle hydrodynamics rendering is also introduced.

2.1 Clinical background

Vascular diseases are the number one cause of death worldwide, with cardiovascular disease alone claiming an estimated 17.5 million deaths in 2005 [9]. An increasingly promising therapy for treating vascular disease is minimally invasive surgery and interventional radiology (IR) procedures. These techniques require high quality images enhanced with contrast medium diffusion, i.e. angiography, and need strong skills in the catheterization technique. It starts being integrated into virtual medical education as more and more training systems are being developed for a high-qualified health care.

2.2.1 Minimally invasive surgery

The minimally invasive approach has revolutionized surgical care, significantly reducing postoperative pain, recovery time, and hospital stays with marked improvements in cosmetic outcome and overall cost-effectiveness [3]. Because of its advantages, this technique has been widely used in general surgery, gynaecology, urology, as well as cardiothoracic surgery. Compared to conventional approaches, patients no longer perceive surgery as a threat to their well-being or their ability to regain their normal life style [3], and thus are more likely to be an alternative for patients in all age levels.

In minimally invasive operations, clinicians manipulate their instruments outside the patient body. Doctors use instruments such as guidewires, catheters and needles, navigating their tip inside the body while looking at images showing their interactions with the patient tissues and organs.

With the growth of robotics and computer techniques, they are now playing an expanding role in assisting the surgeon in several minimally invasive procedures [4]. Guiding-images can be augmented with useful information, boring and fatiguing tasks can be left to robotic systems, and surgeon performance can be enhanced and tracked by intelligent computer systems (figure 2.1). Many new operations are under development in leading universities all around the world, for example, ’i-Snake’ surgical robot will use articulated joints powered by special motors, with multiple sensing mechanisms and imaging tools at its ‘head’, to extend the vision and dexterity of the surgeon, allowing them to navigate difficult and restrictive regions of the body [5].

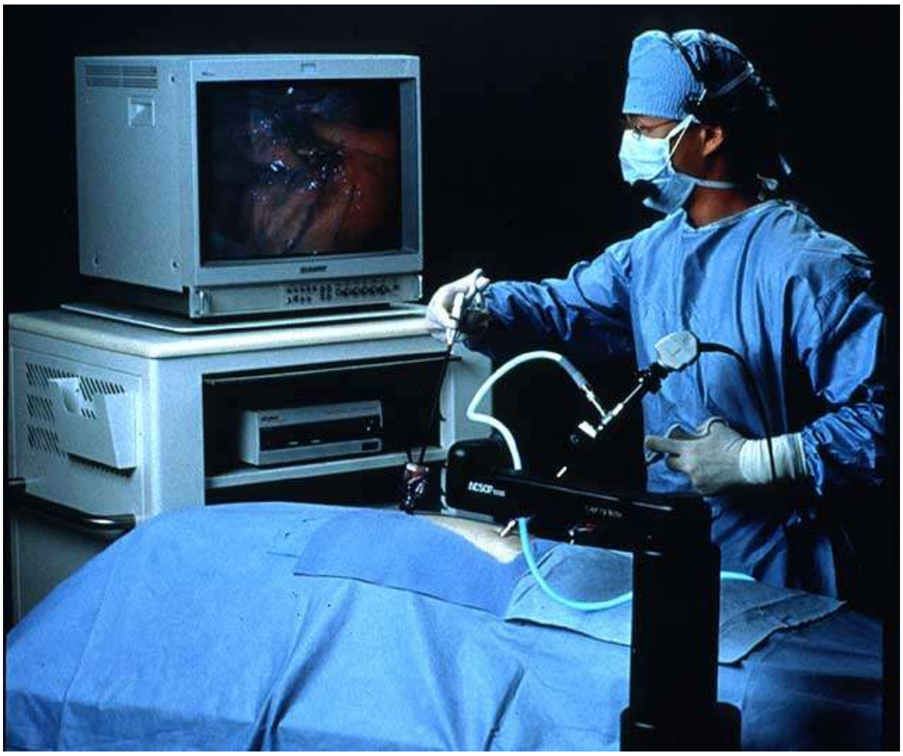


Figure 2.1 A robotic arm that holds the camera and endoscope assembly for the surgeon during an endoscopic procedure [4].

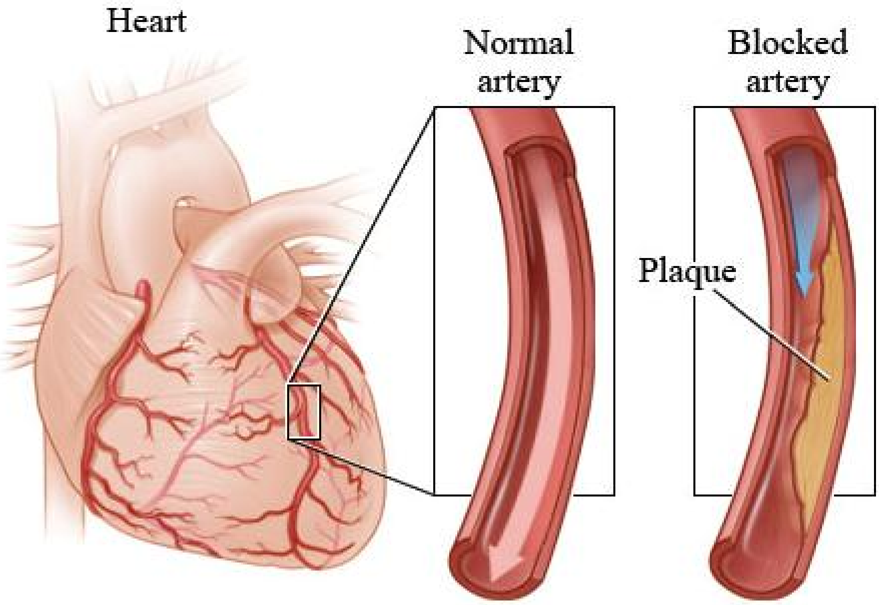


Figure 2.2 Coronary arteries are the blood vessels that supply oxygen-rich blood to the heart muscle. The coronary arteries can become blocked by the build up of plaques. Plaques are made up of extra cholesterol, calcium, and other substances that float in the blood. Over time, plaques can build up on the inside walls of the coronary arteries and block the blood flow. A procedure called angioplasty can open up a blocked artery [47].

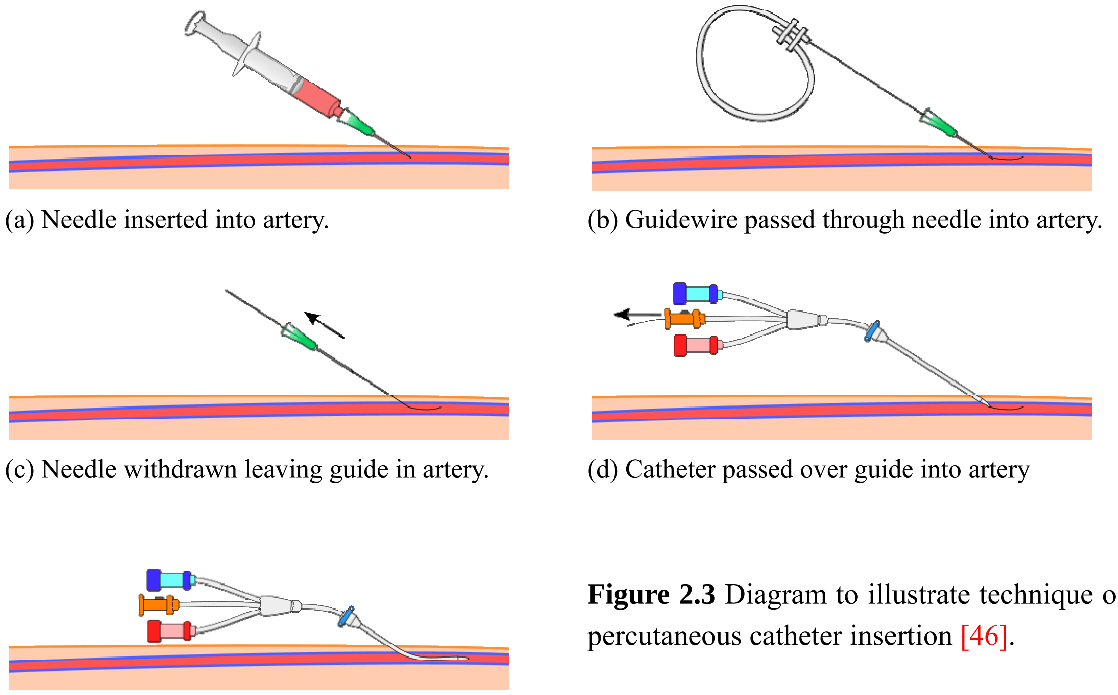
2.2.2 Interventional radiology (IR) and Angiography

Interventional radiology uses image-guided instrument navigation to perform minimal invasive techniques by using specialized tube- or rod-like instruments to access vascular and organ systems through small incisions or natural openings. In recent decades these techniques have been increasingly used for therapy as well as for diagnosis. A wide variety of interventional radiology procedures are used for angiography, treatment of tumours and temporarily occluded arteries, as well as delivery of specific drugs to certain lesion areas, bile drainage and angioplasty (figure 2.2), as well as removal of residual bile duct stones, etc [6 7 8].

The whole process of IR includes methods of percutaneous needle puncture, percutaneous arterial catheterisation (Seldinger technique), instruments navigation, contrast medium injection and medical imaging technique.

- **Needle puncture and Seldinger technique**

The artery is punctured with a needle and a fine malleable guidewire is passed down the needle and into the lumen of the artery. The needle is withdrawn back over the guide and a catheter is then passed along the guide and through the puncture hole into the artery. The guidewire is then used to guide the catheter to any desired level. The catheter can then be pushed along so that its tip lies at the selected place, probably to treat or enhance the visibility of the pathology (figure 2.3) [6]. The tips of these instruments can be designed in order to enter branches of the aorta, providing angiography in some certain areas.



(a) Needle inserted into artery.

(b) Guidewire passed through needle into artery.

(c) Needle withdrawn leaving guide in artery.

(d) Catheter passed over guide into artery

(e) Guidewire withdrawn leaving catheter in artery.

Figure 2.3 Diagram to illustrate technique of percutaneous catheter insertion [46].

- **Contrast medium injection**

When the catheter is arrived at a certain place by manipulation of the instruments, contrast medium can be injected, after removing the guidewire, to obtain X-ray pictures of the interior of blood vessels. By visualizing and measuring flow distributions in the vicinity of a lesion, it is possible to locate narrowing, occlusions, and other vascular abnormalities. Thus contrast medium injection plays a vital role in the assessment of the pre- and post-operative physiological states of the patient.

A medical contrast medium is a substance used to enhance the contrast of structures or fluids within the body in medical imaging. It is commonly used to enhance the visibility of blood vessels and the gastrointestinal tract. Upon injection, the contrast medium is carried by blood flow and circulates through the vascular system until it is eliminated in the kidneys and liver [9 50]. Usual contrast agents are composed of Iomeron-400 or Iotrolan.

In a standard angiography, as mentioned in the previous section, a needle is first inserted in the vasculature, usually in the femoral artery. Then, a guidewire and a catheter are navigated inside the vessels. To visualize the vessels, contrast agent is injected. If the direction of contrast medium injection is different from which of blood flow, it propagates for a short period retrogradely (i.e. towards the heart) along the vessels, including each common iliac arteries at around 6 ml per second, before being mixed with the blood. If the direction is the same, it propagates very fast antrogradely (i.e. towards the feet), and mixing with blood flow rapidly as well.

This process is complex and the resulting angiography pictures are affected by many factors, including initial speed of injection, volume of injection, and vessel bifurcations, as well as vessel cross-sections, blood flow speed and so forth. These factors should be considered as important parameters during the modelling progress, as well as during the simulation strategies.

- **Angiography**

Modern imaging departments use a variety of different techniques to provide images of human internal organs and to demonstrate pathological lesions within them [6]. The basic techniques include X-rays, computed X-ray tomography, and radioisotope scanning, as well as Ultrasound, MRI and so forth. Computed X-ray tomography is generally referred to as computed tomography or CT, while radioisotope scanning is also referred to as nuclear medicine, or radionuclide scanning.

Generally, an angiography is done by taking a continuous series of X-rays while injecting a contrast medium into the vascular structure under examination (Figure 2.4). The contrast medium, usually an iodine solution, provides the density needed for detailed X-ray study of the blood vessels. In interventional radiology, clinicians manipulate the catheterisation mainly based on the generated CT images. Scientists are pursuing clearer image performance for clinicians, and looking for safer (i.e. without side effects) ways for patients.

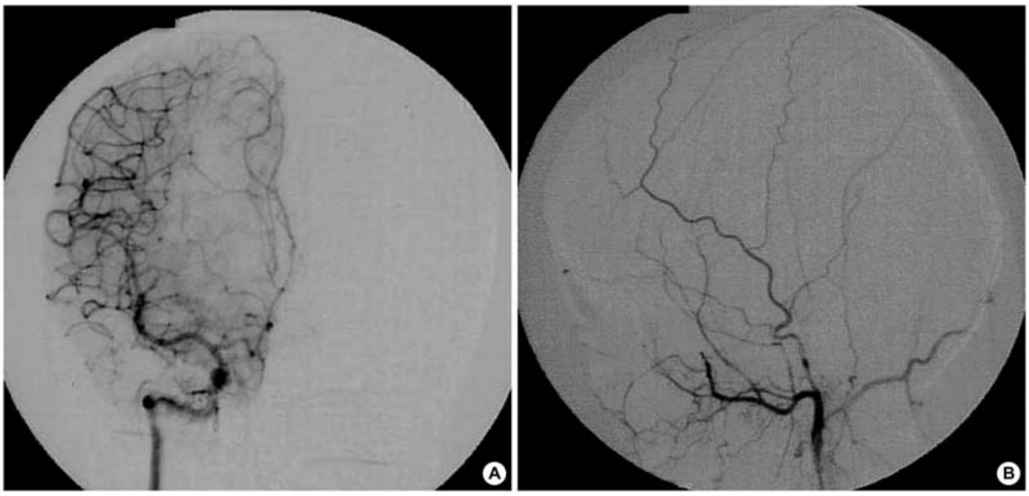


Figure 2.4 Angiography showing the irrigation of the anterior portion of the anterior cerebral arteries on the left side of the brain. No definite abnormal capillary blush or malformed vessels was seen on the right external cerebral artery or internal cerebral artery angiography [48].

Figure 2.5 shows a stenting procedure in interventional radiology, and performs the two angiography pictures before and after propping open arteries blocked with atherosclerotic plaques.

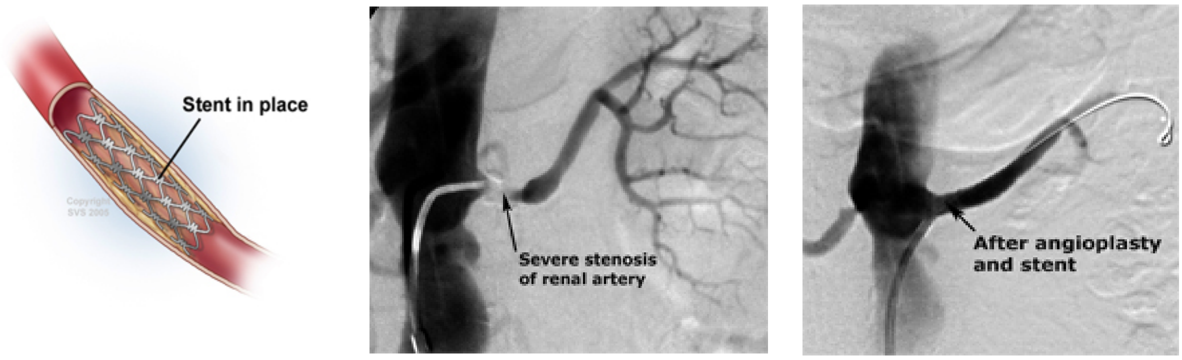


Figure 2.5 A stent is a tiny mesh tube that looks like a small spring, and comes in a variety of sizes, used to prop open arteries blocked with atherosclerotic plaques (Left); Angiography in renal artery with severe stenosis, featuring contrast injection in arterial flow (Middle); and angiography in renal artery after angioplasty and stent (Right).

2.2.3 Medical education and training

Currently, many innovative medical devices are created to deal with more challenging diseases, which require the collaboration between doctors, nurses, radiologists, and even experts from fields such as biomechanics. In order to provide a high-quality health care for patients, education and training inevitably need to bring a medical revolution.

Techniques in virtual reality and simulation have been successfully used in plane pilot education and training. For example, a large number of trainees go to USA each year to experience realistic flying environments, and dealing with different problems, such as training on emergency landing. Further, their flying careers are highly related to the evaluation of the training device. More and more virtual reality technique is translated from aviation to medical education and training. However, compared to flying training, clinical simulation is more complex and needs more accurate and realistic evaluation and performance, especially in some procedures like minimally invasive surgery and interventional radiology.

As mentioned before, the manipulations during interventional radiology are complex. Although with the development of robotic technology and image-guided systems to help radiologists process the manipulation, high-qualified and high skill levels are necessary to ensure a safe intervention and as little trauma as possible for the patient. Traditionally, a five year diagnostic radiology residency program as well as a one year fellowship should be taken before becoming a qualified interventional radiologist as described in [49]. Traditional apprenticeship (“see one, do one”) is followed during this residency, but it has potential risks for the patients.

The advantages of using a simulator compared to normal training on patients are multiple: no risk for the patient, ability to train as much as needed, and capacity to assess the trainee’s skills. Furthermore, for the purpose of training, it is important to have access to a variety of scenarios [9]. The simulator is featuring variable virtual anatomy from the patient datasets, and is able to modify them to design and to simulate different disease scenarios regarding to original datasets (Figure 2.6) [55], which really enrich the training environment. Additionally, users can choose different levels for training according to their current experience.

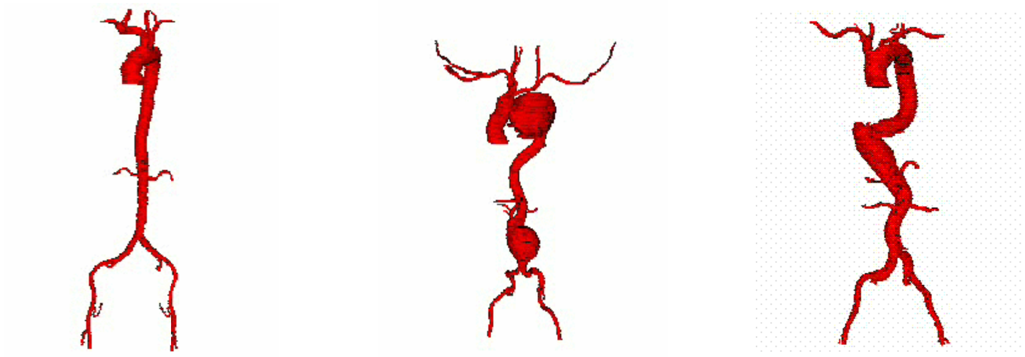


Figure 2.6 Three different vascular anatomies, from femoral arteries (at the bottom) to aorta (in the middle) to neck vessels (at the top). Non pathological aorta (left), two aortic aneurysms (Middle), and aortic dissection (Right) [55].

In the past years, some IR simulators have been developed by research groups all around the word. The IR simulator developed at Kent Ridge Digital Labs uses finite element methods (FEM) to model the interaction between the surgical instruments and vasculature [10 11]. The Simulation Group at CIMIT has developed a suite of simulation tools for IR [10]. It includes an interactive accurate catheter simulation system [9] with a three dimensional vasculature. Moreover, the suite integrates blood flow and contrast medium propagation via a volumetric representation and multi-scale approach. In Stanford SPRING project [13], a surgical simulator was implemented and the realistic simulated blood and irrigation are achieved to enhance the realistic environment using particle hydrodynamics simulation.

Besides, commercial IR simulators are produced by some companies (Figure 2.7), however, their simulation systems lack enough vascular details, thus their realism and immersive level are not enough for a high-qualified training environment. Moreover, there is no vessel deformation provided by the commercial IR simulators.

In this context, a new virtual reality simulator [12] was developed at the Biosurgery and Surgical Technology Group at Imperial College London. It proposes a more realistic model for the instruments, more complex vascular models, a real-time simulation, and the use of an IR specific haptic device [10]. Two virtual instruments, a guidewire and a catheter, are modelled using a mass-spring model [10]. It features a new approach for collision detection, user interaction through haptic devices, and several segmented vascular models with real-time small deformations. The virtual instruments are integrated into the virtual environment for core skills training in vascular interventional radiology as described in [12].

Figure 2.7 Simbionix PROcedure Rehearsal Studio™, an endovascular virtual model training tool. Made by Simbionix Ltd(http://www.simbionix.com/), a medical simulator start-up.

2.2 Simulation background

The simulation involves a standard pipeline including image processing, visualization and graphics rendering. Based on this pipeline, different models are established using physics-based simulation for various kinds of objects. Inspired from X-rays, volume rendering (Figure 2.8) is used mainly in medical visualization, and volume function allows for simulating different absorbance of X-rays. On the other hand, marching cubes (Figure 2.9) techniques provides internal geometric organization of object surface, and supports further use in the resulting geometric environment. Both techniques can be used in simulation rendering, while in interactive system, marching cubes is the preferred method.

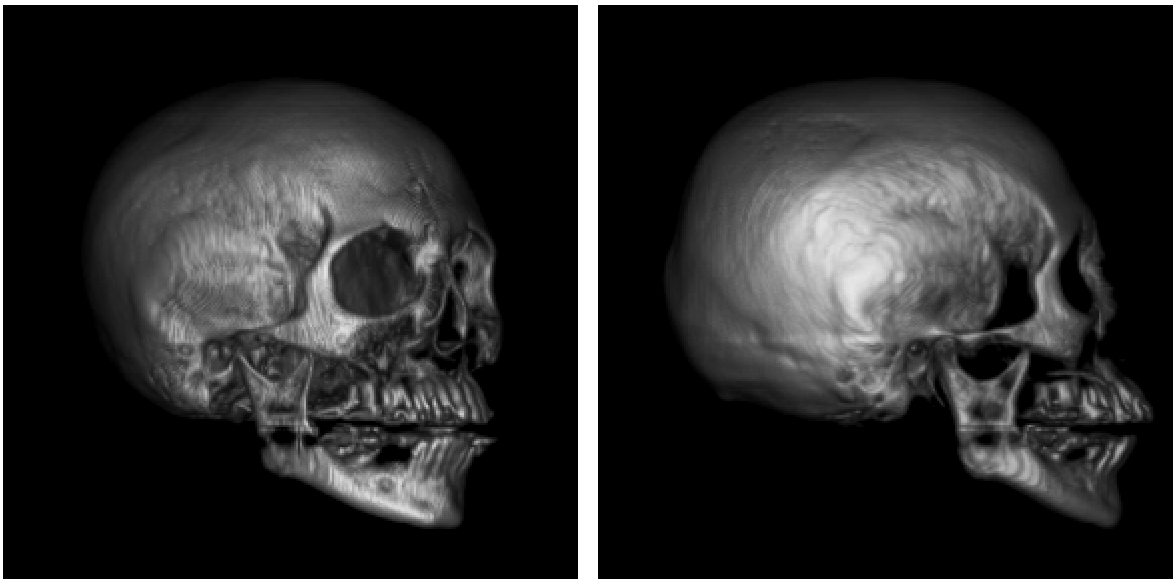


Figure 2.8 Texture based volume rendering of a skull with lighting models and volume functions [52].

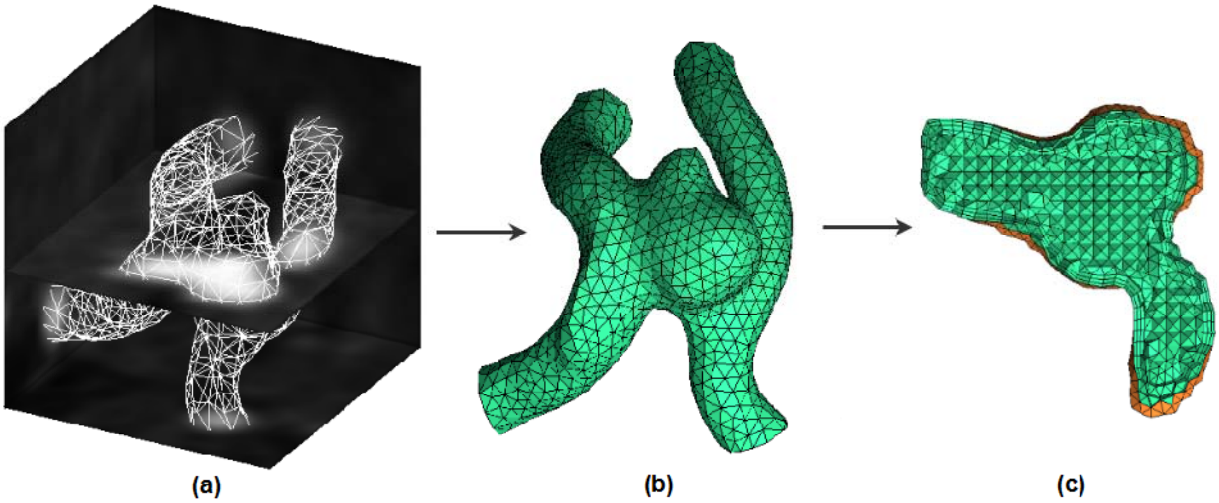


Figure 2.9 (a) Isosurface extracted from volume image using marching cube algorithm; (b) Surface triangular mesh after repairing; (c) Computational grid at the cross-section of aneurysm [51].

2.2.1 Visualization, Image Processing and Graphics

Visualization is a necessary tool to intuitively process the flow of information in today's world of computers. For example, laser scanning systems generate over 500,000 points in a 15 second scan [91]. Without visualization, most of this data would sit unseen on computer disks and tapes [14]. Besides, visualization can extract some important information hidden within the data, thus, it is widely used for medical surgical diagnosis and navigation, as well as virtual medical training simulators. Image processing is the study of two dimensional or three dimensional images. This technique includes transformation, extraction, analysis, and enhancement of images. Computer graphics is the process of 3D geometric reconstruction and creating images using different rendering algorithms (Figure 2.10) [59].

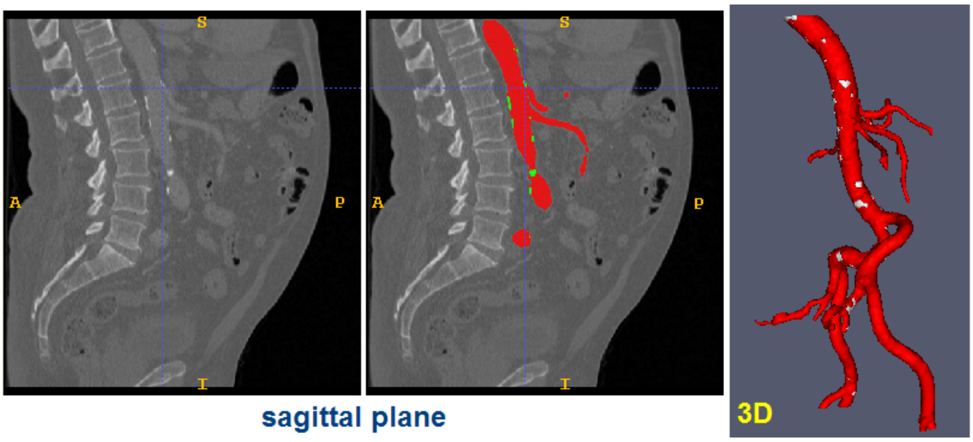


Figure 2.10 Calcification detection from CT image in 2D sagittal view and its 3D reconstruction. Left: original grey image. Middle: artery (in red) and calcification (in green) segmentation results are mapped onto the original image. Right: 3D reconstruction [59].

Therefore, the main steps to achieve virtual image rendering starts by using visualization techniques to get the data from the sources, which can be the output from laser digitizing systems or some CT images. Then, using image processing, the image data is extracted and enhanced. If needed, the image data can be segmented, well organized, and smoothed. After triangulation, a mesh is generated and optimized (Figure 2.10). A practical pipeline about this process using VTK software is described in chapter 4. As the mesh is obtained and optimized, using algorithms such as texture mapping and ray tracing, it can be projected on the 2D screen and completely rendered. Main graphic rendering software includes CFD, OpenGL and Direct X, while VTK (a library using OpenGL) also combines rendering algorithms and can render the images from the mesh representation.

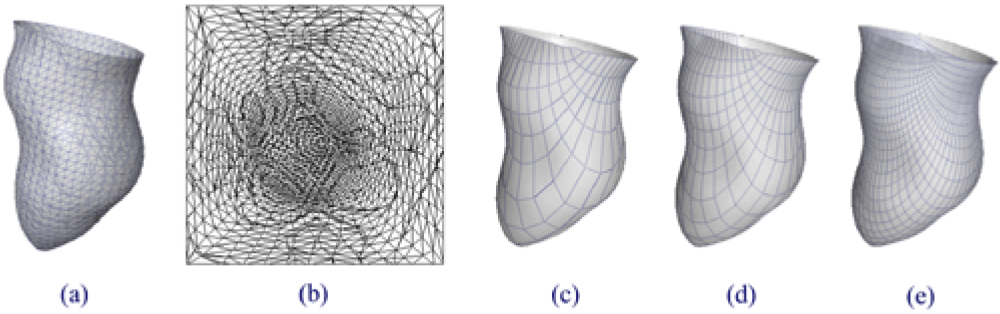


Figure 2.10 (a) Triangulated surface of a left ventricle, (b) Corresponding Harmonic Map on a unit quadrilateral domain and its tensor product B-spline representations at different levels of detail (c-e) [53].

2.2.2 Virtual reality and Haptic device

Virtual Reality (VR) provides a good set of tools to simulate the process of minimally invasive surgery by virtual performance, audio and haptic sensory interaction and a good feeling of immersion. This field contains different disciplines, including computer graphics, visualization, computer vision, robotics, etc. Virtual Reality has many applications in entertainment, education, and scientific visualization. Recently, this technique has been applied to medicine in the context of training, certification and telepresence (Figure 2.11). A VR-based real-time simulation of interventional radiology for angioplasty training, for example the one developed by Duratti et al [56], proposes an environment allowing to carry out the most common procedures: guidewire and catheter navigation, contrast dye injection to visualize the vessels, balloon angioplasty and stent placement.

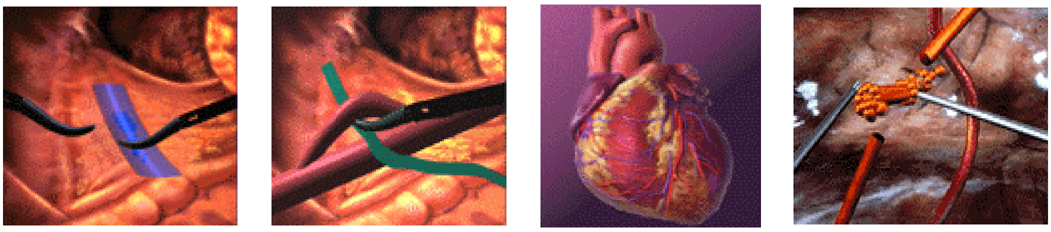


Figure 2.11 Virtual Reality in Surgical Training (Left two pictures); medical education in heart (Middle) and telemedicine (Right)

User interactions are normally allowed in the virtual physical environment. Standard computer input devices, i.e. keyboard and mouse, are often used, but some training systems require different artificial tools as well. Haptic devices are developed for this use. They are mechanical devices that connect the user and the computer. These devices can be designed for needle insertion and catheterisation as described in [10]. In my project, I use the VSP (Vascular Surgical Platform) haptic device (a commercial haptic device for catheterisation, provided by Mentice (http://www.mentice.com/)) (Figure 2.12) to navigate the catheter and guidewire, as well as to inject contrast medium. First of all, the instruments can be navigated to almost everywhere in the vessel system. The full body vascular model simulated, from femoral arteries to neck vessels, consists of over 20,000 triangles, and is optimized for real-time collision detection and visualization of angiograms [9]. Meanwhile, feedback from the simulator always gives an accurate evaluation of operation since even a tiny mistake can be 'felt' by the user through a collision detection process which sends friction information to the VSP. The goal of this project is to allow, during catheterization procedure, to inject contrast medium by the VSP syringe under fluoroscopy.

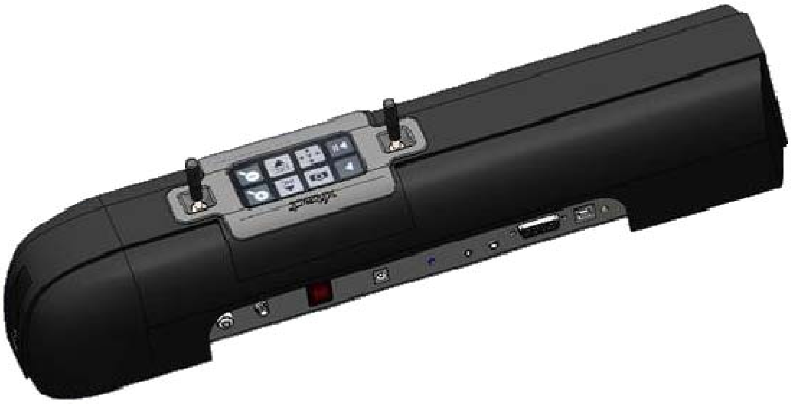


Figure 2.12 VSP Haptic Device

2.2.3 Physics-based simulation

As a research field, physics has a history of thousands of years, investigating about natural phenomenon, such as motion of fluid. Many models have been designed using physics theories for surgical use. For example, a FEM-based model has been developed by Nakao et al for simulating deformation of incision accurately [20]. Mass-spring models have been used for instruments modelling in interventional radiology [10]. In the fluid simulation field, particle-based models and grid-based models are commonly used, and will be discussed in section 2.2.4 in more detail.

Fluids (i.e. liquids and gases) play an important role in everyday life (Figure 2.13). Examples of fluid phenomena are wind, weather, and waves (from ocean waves-to-waves induced by ships or simple pouring of a glass of water) [22]. Though these phenomena look simple and normal, they are difficult and complicated to simulate. Computational Fluid Dynamics (CFD) is a well established research area; however, there are still many challenges, such as blood flow simulation for medical training.

Figure 2.13 A drop disturbs the water's surface, causing ripples to form

Fluids have various complex phenomena, such as convection, diffusion, turbulence, surface tension, viscosity, compressibility and so forth. For a long time, fluid phenomena were typically simulated off-line and then visualized in a second step [22] since the computation was extremely long. However, with the growth of computer techniques and the creation of GPU, many real-time simulations appeared in computer games, but also medical simulators and other virtual environment (Figure 2.14).

Figure 2.14 Blue natural water Simulation (Left); Concept house (http://www.concepthouse.com/) company products - Touch simulation (Middle) [23]; Green smoke simulation (Right)

Many great theorems, such as Newton's Law, explain fluid laws in more or less detail. With the growth of the CPU and GPU of the computers, those physical situations can now be simulated and computed quickly and dynamically. Computer-assisted systems based on Virtual Reality techniques are widely utilized as essential tools for surgical planning and for evaluation of surgical intervention [17 18 19], and are also successfully used in realistic simulators with the help of haptic devices.

2.2.4 Fluid Modelling Simulation

Computational Fluid Dynamics research has a long history. The famous Navier-Stokes equation (given in next sub-section) was formulated by Claude Navier in 1822 and George Stokes in 1845. It describes the dynamics of fluids and conservation of momentum. Besides, these two equations, a continuity equation describing mass conservation and a state equation describing energy conservation, are all meaningful to fluid simulation. Moreover, following Einstein's dictum, "everything should be made as simple as possible, but not simpler.", computers are able to simulate fluid using Navier Stokes equations in a smooth way and in real-time if they can approach the equations by simple mathematical models and algorithms.

In the last thirty years, fluid simulation techniques have been developed in the context of Computer Graphics (CG) for several purposes. In 1983, Reeves [24] introduced particle systems as a technique for modelling a class of fuzzy objects. Since then, different methods of fluid simulations have been created: the particle-based Lagrangian approach, the grid-based Eulerian approach, Lattice Boltzmann simulation approach, and procedural simulation methods, as well as height field representation, linked volume-based rendering, and so forth. In the area of computer games (Figure 2.15), many game physical engines with realistic fluid effects are developed. In medical virtual reality, one successful application was developed by the Simulation Group at CIMIT (http://www.medicalsim.org/), which simulates blood flow and contrast medium propagation via a volumetric approach [9] in a suite of simulation tools for Interventional Radiology.

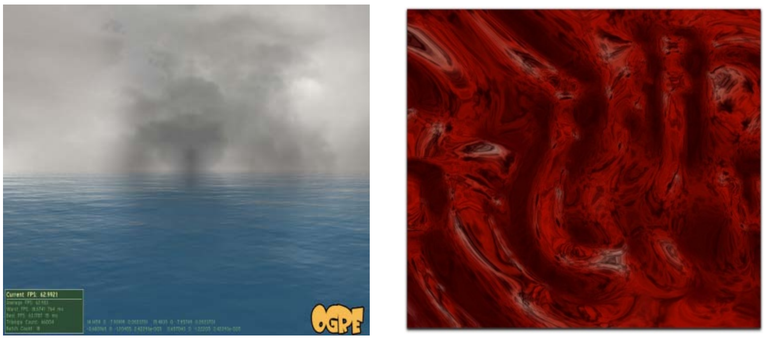


Figure 2.15 Water and Smoke Simulation using Ogre Game Engine [18], Concepthouse company products – Blood work simulation (Right) [17]

In this work, basic fluid mechanics background and many fluid rendering methods are presented. Volume rendering [9], linked volume simulation and particle-based simulation methods will be explained in detail since they are relevant to blood flow and contrast medium propagation simulation. The most suitable way to simulate angiogram rendering with a high level of fidelity will also be discussed and used in the project.

2.2.4.1 Fluid mechanics background

Fluid mechanics is the study of how fluids move in Nature and what forces influence them. It can be divided into fluid statics (motionless fluids), and fluid dynamics (fluids in motion). Blood flow is well modelled by solving Navier-Stokes equation in [9]. For simulating contrast medium propagation, both signal-enhanced curve model and advection diffusion model are introduced.

- **Navier-Stokes Equations**

In this research project, flow distribution in the vascular network is more relevant when identifying and quantifying vessel pathology than local fluid dynamic pattern [9]. Navier-Stokes equation (N-S) (Equation 2.1) is adequate for our real-time catheterization simulator [12]. Blood flow in each vessel can be modelled as an incompressible viscous fluid flow through a cylindrical pipe. As shown in SIGGRAPH 2007 Fluid Simulation Course Notes [25], blood flow through the vessels can be modelled using the following equation:

$$\rho\,(\frac{\partial u}{\partial t} + u \cdot \nabla u) = \rho g - \nabla P + \mu \nabla \cdot \nabla u \qquad \textbf{(2.1)}$$

Where u is the velocity of the fluid, it can be considered as a function of t, $u = u(\mathrm{t})$. ρ stands for the density of the fluid. For water, this is roughly 1000kg/m3 and for air this is roughly 1.3kg/m3. P stands for "pressure", the force per unit area that the fluid exerts on anything. g is the familiar acceleration due to gravity, usually (0, −9.81, 0) m/s^2. μ is technically called the "kinematic viscosity". It measures how viscous the fluid is. Fluids like molasses have high viscosity, and fluids like alcohol have low viscosity: it measures how much the fluid resists deforming while it flows (or more intuitively, how difficult it is to stir) [25].

Using equation 2.1, the time function of velocity field $\mathrm{v}(t)$ is obtained by incrementing by Δt from the time t, from v(t) to $\mathrm{v}(\mathrm{t} + \Delta t)$. The simulation should include self-back advection, diffusion modification, external forces and vorticity acceleration, and final projection. Where advection describes how the fluid carries itself around, diffusion describes how the fluid motion is damped [26], and vorticity refinement could show the feature of high detail turbulent structure, while final projection can finish the last step of simulation.

- **Poiseuille Law**

In fluid dynamics, the Poiseuille equation is a physical law that gives the pressure drop in fluid flowing through a long cylindrical pipe. This law is the first order form of Navier-Stokes equations, and it is widely used in a fluid environment with almost constant speed. The assumptions of the equation are that the flow is laminar viscous and incompressible and the flow is through a constant circular cross-section that is significantly longer than its diameter [21].

The standard fluid dynamic notation is in the following equation:

$$\Delta P = -\frac{8\mu \mathrm{LQ}}{\pi \mathrm{r}^4} \qquad \textbf{(2.2)}$$

Where ΔP is the pressure drop, L is the length of pipe, μ is the dynamic viscosity, Q is the volumetric flow rate, r is the radius, π is the mathematical constant (approximately 3.141592654). In this project, Poiseuille law is used as a basic theory for contrast medium fluid speed calculation, regarding to the physics equation:

$$\Phi = \frac{\mathrm{d}V}{\mathrm{d}t} = v\pi R^2 = -\frac{\pi R^4}{8\eta}\left(\frac{-\Delta P}{\Delta x}\right) = \frac{\pi R^4}{8\eta}\frac{|\Delta P|}{L} \qquad \textbf{(2.3)}$$

Where Φ is the volumetric flow rate, V is a volume of the liquid poured (cubic meters), t is the time, *v* is mean fluid velocity along the length of the tube, *x* is a distance in direction of flow, *R* is the internal radius of the tube, ΔP is the pressure difference between the two ends, η is the dynamic fluid viscosity, and *L* is the total length of the tube in the x direction [21].

- **Signal-Enhanced Curve Model**

The use of contrast medium has a number of benefits for vascular studies [27]. The first model was developed by Maki et al. by assuming a slowly moving flow in a straight pipe [28]. The contract medium is assumed to move along in the pipe as a straight line [27]. Then, the signal is formulated and simplified to a mathematical function of the velocity and the other function of contrast transport and diffusion, which can be considered as the measured signal-enhanced curve. This model, as the signal-enhanced curve, implies the history of distribution of contrast medium, which will be considered and tried in the project.

- **Advection-Diffusion Model**

According to the simulation of contrast medium propagation, an advection-diffusion model for contrast medium propagation was used by Wu et al, which describes the distribution of contrast medium concentrations X(*x, d, t*) as a function of curvilinear coordinates *x* along the centreline of a vessel, the distance d to the centreline, and time t (Equation 2.4) [9]:

$$\frac{\partial C(x,d,t)}{\partial t} + u(x,d,t)\frac{\partial C(x,d,t)}{\partial x} + KC \cdot \nabla \cdot \nabla C(x,d,t) = I(x,t) \qquad \textbf{(2.4)}$$

Where *I*(*x*, *t*) is the injection rate of contrast medium, KC is the contrast medium diffusion factor, and $\mu(x, d, t)$ is the laminar flow velocity along the axial direction of each vessel [9]. This velocity can be modelled as a parabolic profile (Equation 2.5).

$$u(x,d,t) = \frac{1}{4\eta}\frac{\Delta P}{\Delta x}(r^2 - d^2) = \frac{2}{\pi r^2}Q\left(1 - \frac{d^2}{r^2}\right) \qquad \textbf{(2.5)}$$

Where Q is the vessel flow rate, which can be calculated by Poiseuille Law (Equation 2.3), which also comes from the Navier-Stokes Equation (Equation 2.1), Q is given by:

$$Q = \frac{\Delta P}{R} \quad with \quad R = \frac{8\eta L}{\pi r^4} \qquad \textbf{(2.6)}$$

Where ΔP is the pressure drop, η is blood viscosity, *r* is vessel radius, and *L* is vessel length. Compared to most previous works in which only concentration at the centreline was modelled, this model provides two important flow features [9]. One is that the propagation front profile is not flat, and due to the low velocity near vessel walls, some contrast medium can remain longer time in vessels. The other feature is that in the Equation 2.6, using the square of the radial distance can maintain a parabolic profile, which is reasonable since depending on the radial position. It is not important for a precise variation as enough interpolation exists between the border value and at the centre value.

The previous subsections presented basic fluid mechanic. The next sections describe different models solving and using these equations.

2.2.4.2 Grid-based simulating fluid

Usually, Navier–Stokes equation (Equation 2.1) is discretised on regular or irregular grids using the finite difference method, and various numerical methods could be used to resolve the linear systems and the temporal integration of the velocity [29]. Foster et al [30] were the first to solve the three-dimension fluids using Navier-Stokes equation. And in 1999, Stam [31] introduced the semi-Lagrangian method to evolve the non-linear advection term using this N-S equation, and who made the simulation more stable even for large time steps [29]. Under Stam's framework, many researchers develop different models to solve N-S equation for different purposes and get different realistic simulations.

In Stam's simulation framework, each quantity is defined on either a two-dimensional or three-dimensional grid. The basic structure of grid includes origin O, length of each side *L*, and number of cells in each coordinate *N*. Thus, the length of each voxel can be formulated as the equation:

$$Lv = \frac{L}{N} \tag{2.7}$$

The velocity field is defined at the centre of each cell as shown in Figure 2.16. First of all, two grids for each component of the velocity are created. Then, at each time step, one of the two grids corresponds to the solution obtained in the previous step, thus the new solution can be stored in the second grid. After each time step, the grids are swapped. Similarly, any scalar field corresponding to a substance transformed by the flow can be modelled using this method. Besides, the specific value of boundaries of these scalar fields should be considered as the periodic or fixed boundaries, which has a detailed description in [31 36].

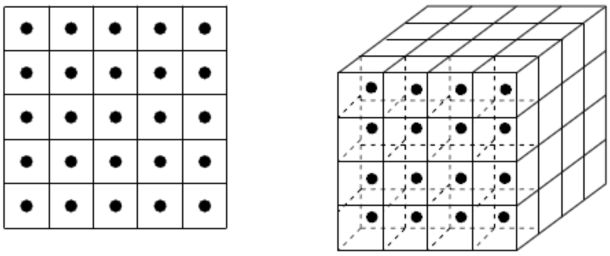


Figure 2.16 The values of the discretised fields are defined at the centre of the grid cells

Using this method with advecting solid textures, interactive fluid simulation can be achieved. However, disadvantages of using this method in the real-time dynamic simulation, includes poor scalability and lack of control. It can be addressed by assuming that stationary obstacles are aligned with grid cells during the finite difference discretisation, and by appending terms to the N-S equations to include a forcing function, which is described in [30].

2.2.4.3 Lattice Boltzmann simulating fluid

In the 1980s, CFD researchers applied the idea of cellular automata to the Boltzmann equation, and led to the modern Lattice Boltzmann methods (LBM) [29]. The lattice Boltzmann formulation is also a grid-based method, but originated from the field of statistical physics [32].

Following the basic idea of grid-based rendering as described in section 2.2.4.2, the basic algorithm of Lattice Boltzmann includes two steps, the stream step, and the collide-step [32]. Using a LBM, the particle movement is restricted to a limited number of directions. Here, a three-dimensional model with 19 velocities (commonly denoted as D3Q19) will be used. The velocity vectors take the following values:

$e_1 = (0, 0, 0)T$, $e_{2,3} = (\pm1, 0, 0)T$, $e_{4,5} = (0, \pm1, 0)T$, $e_{6,7} = (0, 0, \pm1)T$, $e_{8..11} = (\pm1, \pm1, 0)T$, $e_{12..15} = (0, \pm1, \pm1)T$, and $e_{16..19} = (\pm1, 0, \pm1)T$.

For each of the velocities, a floating point numbered f1..19, needs to be stored, representing a blob of fluid moving with this velocity. Each blob is thought to be a collection of molecules or particles. These two-dimensional and three-dimensional models depend on particle distribution functions (DFs). Thus, in the D3Q19 model (Figure 2.17), the DFs of length 0 means the corresponding particle never moves, DFs of length 1 means the corresponding particle moves at speed 1, and DFs of length 2 means moving speed is $\sqrt{2}$.

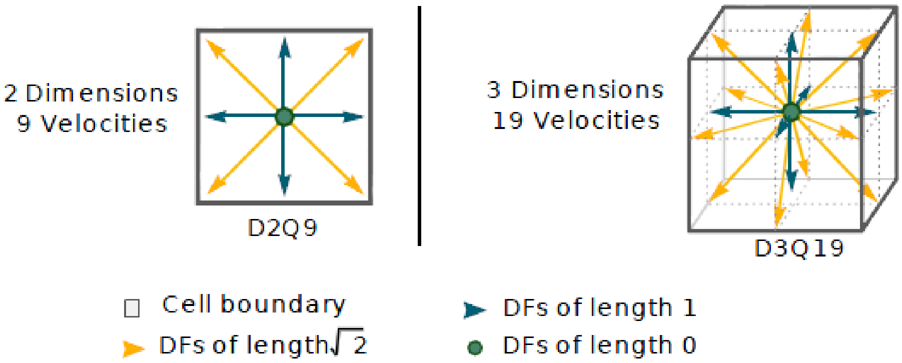
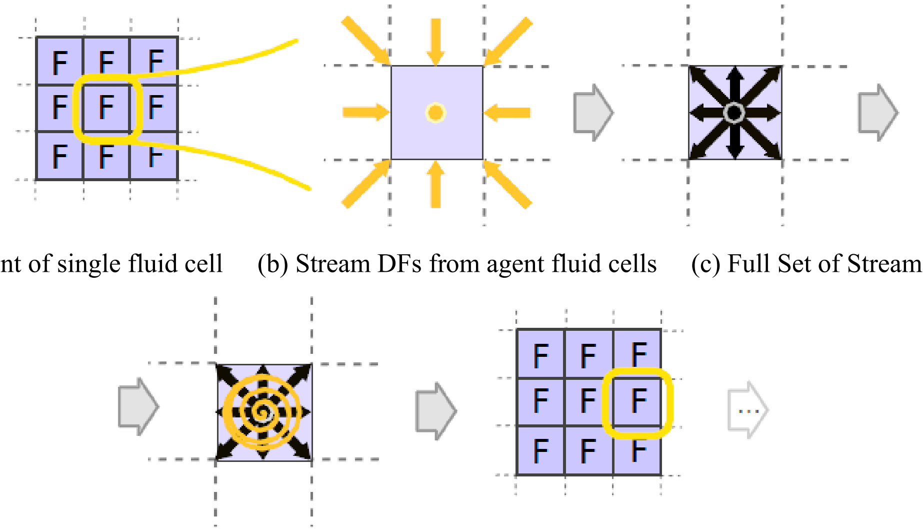


Figure 2.17 The most commonly used LBM models in two and three dimensions [32]

During the stream step, all DFs are advected with their respective velocities. This propagation results in a movement of the floating point values to the neighbouring cells. However, the stream step alone is clearly not enough to simulate the behaviour of incompressible fluids, the particles for which collisions happens. Thus the main task in collision step is to weight the DFs of a cell with equilibrium distribution functions [32], and these functions depend solely on the density and velocity of the fluid. The process of the two steps is shown in Figure 2.18.

Moreover, the situation of a fluid cell next to an obstacle can also be simulated by a two step LBM, which has additional ability to form a turbulent fluid model. A good explanation is described in [32]. Compared to other methods, LBM has the advantages of being a nice parallel process, handling complex boundaries, and good programmability. However, as the LBM performs an explicit time step, the allowed velocities in the simulation are typically limited to ensure stability [29].



(a) Treatment of single fluid cell (b) Stream DFs from agent fluid cells (c) Full Set of Streamed DFs

(d) Compute Velocity and Density Collide streamed DFs (e) Store DFs in Target Grid, continue with next Cell

Figure 2.18 Overview of the stream and collide steps for a fluid cell [32]

2.2.4.4 Procedural fluid simulation

A procedural method animates a physical phenomenon directly instead of simulating its cause [33, 34]. For example, Long et al [35] show that slipping boundary conditions can be imposed by solving the mass conservation step using cosine and sine transforms instead of the Fourier transform (Figure 2.19).

Procedural simulation has a wide use in procedurally generating the surface of a lake or ocean by superimposing sine waves of a variety of amplitudes and directions. An advantage of procedural animation is its controllability, an important feature in games. The main disadvantage of simulating water procedurally is the difficulty of modelling correctly the interaction of the water with the boundary or with immersed bodies [25].

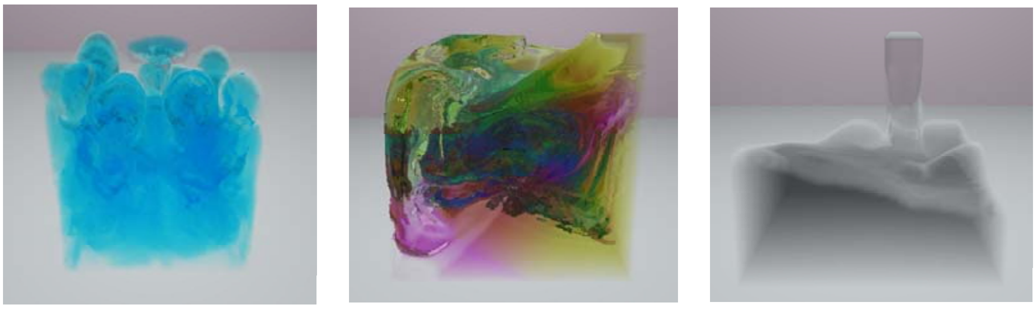


Figure 2.19 Rayleigh-Taylor instability simulation (Left); a multi-liquid simulation (Middle); pouring water (Right) [35]

2.2.4.5 Volumetric representation

Volume rendering is typically used to generate images that represent an entire 3D dataset in a 2D image. There are many related techniques to produce an image, including direct rendering techniques, geometric primitive rendering techniques, or a combination of these two methods (Figure 2.20) [37].

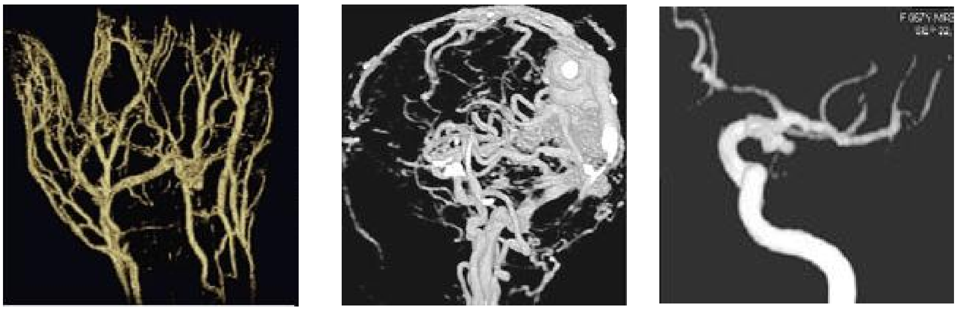


Figure 2.20 Volume renderings of contrast-enhanced magnetic resonance angiography of the hand (Left); Contrast-enhanced magnetic resonance angiogram of an arterio-venous malformation (Middle); Three-dimensional time-of-flight magnetic resonance angiography of an aneurysm at the origin of a foetal posterior cerebral artery (Right) [38].

To visualize the angiogram in three-dimensions, a contrast medium propagation model can use 1D advection-diffusion (Equation 2.4) as discussed in section 2.2.4.1, and previously prescribed in [9], which use volumetric representation to create the lumen of vessels through the way of dividing the interior space into equal sized voxel elements. By using advection-diffusion model, each voxel can be related with a curvilinear location x along each vessel. Then, the contrast medium concentration value of a sample point along a vessel is mapped to the voxels in the vicinity [9]. More than 40 million equal sized voxels with size 0.25×0.25×0.25mm^3 was generated in [9]. It was set to very detail vascular models with radius of vessel from 0.5mm to 17mm. However, it can no longer be rendered in real-time because of the large amount of generated voxels.

2.2.4.6 Multi-Scale approach

To maintain both high-resolution visualization and real-time rendering [9], an efficient rendering technique such as the marching-cubes [Lorensen87] in 3D is considered. The basic assumption of the technique is that a contour can only pass through a cell in a finite number of ways. A vertex is considered inside a contour if its scalar value is larger than the scalar value of the contour line, while vertices with scalar values less than the contour value are said to be outside the contour [14]. In 3D space, a voxel has eight vertices and each vertex can be either inside or outside the contour. There are $2^8 = 256$ possible ways that the contour passes through the cell. However, 256 possible cases have been reduced to 15 cases by reflections and symmetrical rotations [40].

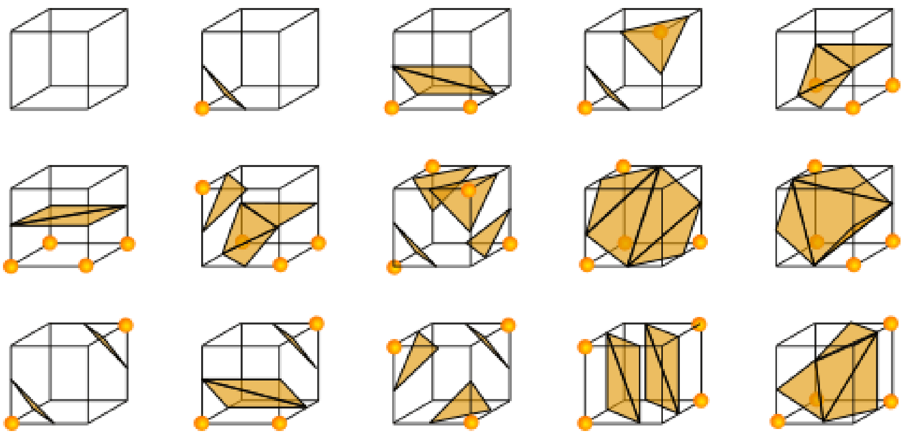


Figure 2.21 The 15 possible cases of the marching-cube [39]

Similar to the marching-cube (Figure 2.21), a multi-scale approach based on subdivision [9] was developed to achieve an efficient high-resolution visualization and real-time rendering. The result of this method leads to the generation of only 9.2 million multi-scale voxels under the situation of the section 2.2.4.5, where 40 million voxels should be generated. The start of this method is a prescribed initial grid size; each voxel is tested and labelled as internal, boundary, and external [9]. Then, if a voxel is regarded as internal, it can be immediately stored, while the boundary voxels should be subdivided into eight equal sized sub-voxels. Interactively, according to the set of surface polygons intersected by the parent voxel, each sub-voxel is examined and labelled. If a sub-voxel missed the intersection with the surface then it can be labelled as inside the surface and stored. If a sub-voxel is still labelled as boundary voxel then it needs further subdivision into eight smaller particles. A return mechanism can be predefined as the number of subdivisions, which can control the granularity under a certain situation.

Image quality in contrast-enhanced magnetic resonance angiography is governed by the timing of contrast injection and data acquisition [15]. Thus, different sized particles have different amount of radiation attenuation under fluoroscopy [9]. To achieve the qualified image under transportation and propagation of contrast medium, each particle’s rendering size can be linearly adjusted, and its intensity can be exponentially modified according to its dimension. Thus, using the efficient multi-scale approach, the aim of real-time propagation simulation can be achieved. Besides, this approach is implemented using a programmable shader and runs at interactive frame rates [9].

However, this method would be time-prohibitive when computing a volume-based diffusion as the contrast propagates into invisible capillary vessels. Indeed, in these small vessels, a blush must be simulated since it is a very important visual cue which can help the physicians to realize the concentration of contrast material and therefore the concentration of vessels in a certain area [41].

2.2.4.7 GPU acceleration fluid simulation

Modern GPUs are very efficient at manipulating and displaying computer graphics. Their highly parallel structure makes them more effective than general-purpose CPUs for a range of complex algorithms [42]. To render real-time blush under the propagation of contrast medium, GPU rendering is used to compute an image-based diffusion.

The basic idea is that when the contrast medium propagates to invisible capillaries in the blushing area, first are rendered the multi-scale particles using multi-scale approach as mentioned in section 2.2.4.6 and then it uses a two-pass (horizontal and vertical) convolution shader to compute a Gaussian blur [9]. The shader technique is fully computed through GPU. Moreover, the blur radius can be modified to get a high-level visual coherence according to the real angiogram, which can be produced as a blur image.

The three images (blur image, multi-scale particle image and simulated X-ray images by volume rendering) are combined together to form the final angiography (Figure. 2.22).

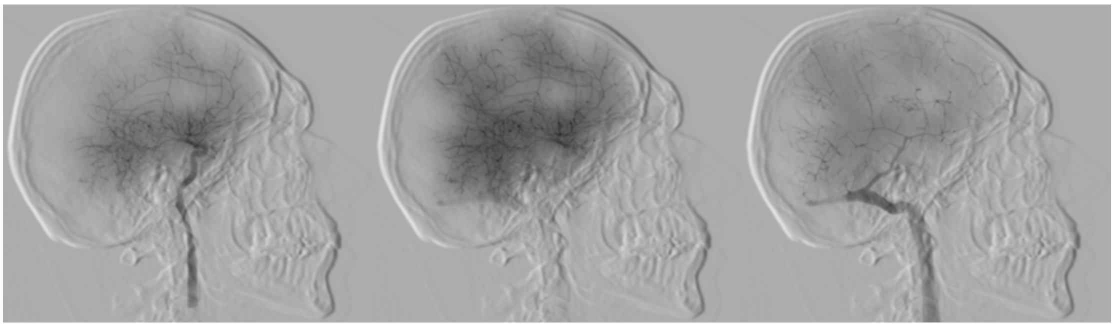


Figure 2.22 High-fidelity real-time simulation of angiography in the brain, featuring contrast injection in arterial flow (Left); blush (Middle); and transition to venous side (Right) [9].

2.2.4.8 Linked volume-based simulating fluid

Using volume rendering, a realistic fluid simulation can be achieved as mentioned in section 2.2.4.5 and section 2.2.4.6. Unfortunately, it cannot fully simulate all the behaviour of fluid propagation if not using the GPU acceleration described in section 2.2.4.7. Thus, other approaches are given in order to look for a qualified, fast and effective fluid simulation.

Linked volume-based method is related to volume rendering, and is extended from basic mass-spring method [43]. It is based on a discretisation of the entire object volume into same-sized spaced elements (Figure 2.23a). The elements are connected with their neighbours in space by resorting to springs and dampers. Using this method, a rapid propagation can be achieved, and almost all the mechanical behaviour can be simulated. Moreover, any complicated interactions can be simulated by just adding or cutting some related joint elements (Figure 2.23b).

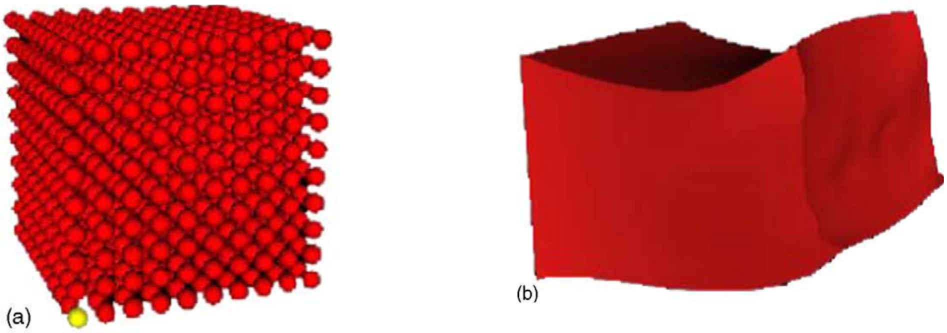


Figure 2.23 Deformation of a cube discretised into linked volumes: **(a)** undeformed system; **(b)** deformed system, rendered with the help of an additional boundary representation of the surface [39].

2.2.4.9 Chain mail approach

The chain mail algorithm is relatively close to the volumetric one. It is described in [44]. Similar to the motion of molecule in the fluid, within a certain limit, each link of the molecule can move freely without influencing its neighbours, while major change is propagated directly to the corresponding adjacent links (Figure 2.24).

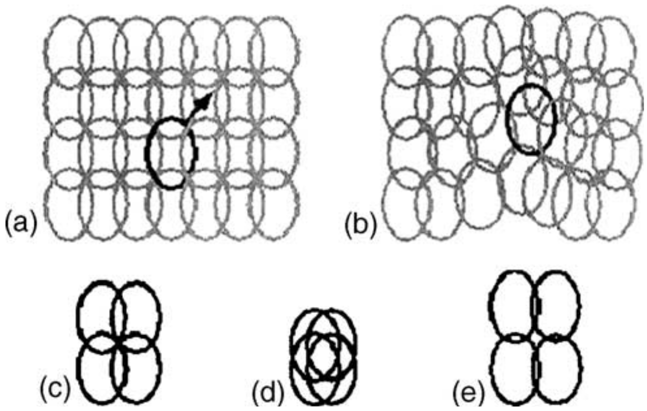


Figure 2.24 Principle of chain mail algorithm [44]: deformation of a 2D mesh when moving a link in the direction of the arrow: **(a)** initial state; **(b)** deformed state: the neighbouring links are displaced to maintain maximum and minimum distances. Set of links: **(c)** initial state; **(d)** maximum compression; **(e)** maximum stretching [42].

The advantages of using volume linked method and chain mail method are their simplicity, and the possibility to achieve complex interactions. However, since there is no force between the links, it is hard to model the viscosity of the fluid, moreover, a very high price is paid if improving the volumetric character and allowing for more complex mechanical behaviour [42].

2.2.4.10 Particle-based simulating fluid

Compared with three-dimensional grid fluid simulation, particle-based methods, such as smoothed particle hydrodynamics, do not use mass conservation equations and convection terms. It thus reduces the complexity of the simulation (Figure 2.25) [22]. Besides, the particles can be used to simulate the surface of the fluid directly, which is better than using linked volume methods and their cutting and smoothing methods. By using point splattering and marching cubes-based surface reconstruction (mentioned in section 2.2.4.6), this method is fast enough to be used in real-time interactive system, and thus allows users to interact with the model it simulates.

Particle-based rendering can be considered into three different situations, including simple particle systems, particle-particle interactions and smooth particle hydrodynamics, which can be solved by using N-S equation and generate a continuous and smooth field.

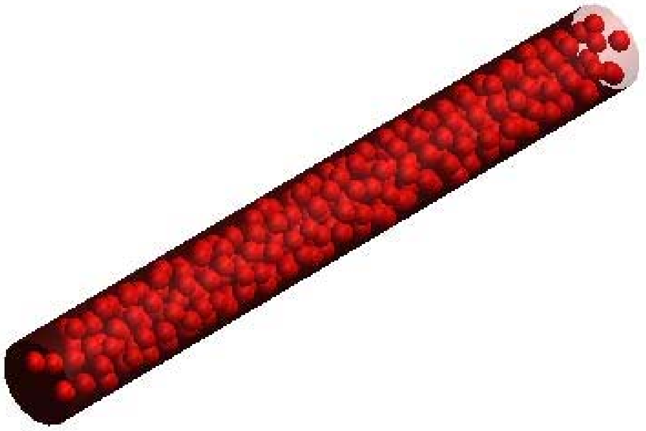


Figure 2.25 Flow of blood particles through a cylindrical model of an artery

- **Simple Particle System**

A simple particle system is a particle system without particle-particle interaction. Such system can be simulated rapidly and a large number of particles can be simulated in real-time. The properties of a particle are its mass m, its position x, its velocity v and the accumulated external force f. A particle can be created with meaningful position and velocity at the beginning of the simulation. Similarly, a large number of particles can be created with positions and velocities and then generated during the simulation by emitting them with a certain rate (particle per second) and a velocity distribution. Besides, lifetime of a particle is given and a particle will be faded out smoothly when reaching that time.

- **Particle-Particle Interaction**

Naturally, in a blood flow simulation with particles, it is essential that the particles interact together to create the impression of a viscous fluid. The interaction force acts along the connection between the particles, however, the computing cost is huge. For example, with N particles, there are O(N^2) possible interactions [25]. That means with 1,000 particles 1,000,000 interactions should be considered in real-time. To reduce the complexity, a distance d is introduced. Over this distance, the particles do not interact with each other. At each time step, the particles can be considered as being filled into a regular grid with cell spacing d and in each cell, the number of particles is a constant. Thus, the particle-particle interactions are simulated in the same or adjacent cells and a real-time particle-particle interactive simulation can be achieved.

- **Smooth Particle hydrodynamics**

By using the previous supposed situation, a basic particle-based fluid simulation can be achieved. However, for medical training purposes, the effect and realism is not satisfactory. Thus, a smooth particle hydrodynamics method is introduced which considers pressure, external forces and viscosity by using the N-S equation. Moreover, to achieve a smooth continuous field from particles' quantities, the normalized smoothing kernel is defined to be symmetric around a particle using kernel function [25]. Thus, we can get all ingredients from the smoothing kernel to compute a smooth density field from the individual positions and masses of the particles. In other words, we can get the density values of the individual particles. Using the density values, the smoothed field can also be calculated mathematically.

In the grid-based formulation, fluids are described by a velocity field, a density field and a pressure field. The evolution of these quantities can be given by solving two equations: conservation of mass and conservation of momentum (N-S Equation). But, if using particle-based formulation, the two equations can be simplified to the following equation: and the description of the reason is introduced in the section 2.2.5.

$$\rho \, \frac{\partial u}{\partial t} = \rho g - \nabla P + \mu \, \nabla \cdot \nabla u \qquad \textbf{(2.8)}$$

On the right hand side of the N-S equation (Equation 2.8) there are three terms: pressure, external forces and viscosity. The sum of the three terms determines the change of momentum which can be easily calculated by the $\partial u / \partial \mathrm{t}$ since the substantial derivative of the velocity field is simply the time derivative of the velocity of the particles [25]. Thus, the acceleration of a particle (the time derivative of the velocity of a particle) equals to the body force (sum of forces of pressure, viscosity and external forces) over density field (a density value around the particle calculated by normalized smoothing kernel, individual positions and masses of the particles). Using fluid mechanics symmetric equation and ideal gas state equation, pressure force can be solved. Using symmetric velocity difference equation [25], the force of velocity can be computed. It can be solved by using only the equation of simple particle systems since the external forces, such as gravity and collision forces, can be directly applied to the particles.

The major advantage of using particle-based rendering is that it needs less memory and the simulation model is comparatively simplified. Moreover, the particles can be used to simulate directly the surface of the fluid, and there is no need to use other cutting and smoothing methods. Because of these advantages, rendering a particle-based fluid can reach a real-time effect and can also allow user interactions under the current capability of computing.

However, an important disadvantage is that it is difficult to represent a smooth surface with particles [45]. And it is also a difficult task to robustly handle surface tension effect for small features. Further, it is costly for particles to find their nearest neighbour, and it is also complicated for dealing with complex moving boundaries. But, these problems can be modified by different new solutions. For example, Muller et al created a new way to model surface tension forces basing on ideas of Morris [50]. Thus, in this project, I propose a particle-based approach based on Smoothed Particle Hydrodynamics to animate the blood flow and contrast medium propagation. It will be the method I will implement during my project.

In basic physics, the interactive forces can only change the internal momentum of particles, while the total momentum of the particles remains the same. This theory is called conservation of momentum, which describes how the fluid accelerates due to the forces acting on it [25]. The representation of the momentum is $\partial u / \partial t$, where u is the 3D velocity, and ∂t is time step. By using particles to simulate the physical fluid property, the momentum conservation equation can be more briefly described and therefore computationally more efficient.

Wu et al [9] modelled the flowing fluid of blood by 1D laminar flow model. It is based on two assumptions: first, eliminating the turbulent flow which is rarely observed in interventional radiology procedures; second, suppose that the blood flow is incompressible. In my project, these assumptions are also used, but time permitting, more complex forms of blood flow in virtual angiograms will also be considered. At first, no blood will be simulated, only contrast medium with constant speed and no attenuation. If real time can be achieved with this highly simplified model, the complexity will be increased: adding first different contrast medium speeds, attenuation of the density of the contrast medium, and finally interaction with the blood flow.

Compared with Wu's work [9], the main method chosen to simulate blood flow and contrast medium propagation will use particle-based rendering. A particle can be considered as a little blob of blood or contrast medium, and it has a mass m, a volume V, and a velocity u, thus, the acceleration a can be derived from Newton's second law:

$$a = \frac{\partial u}{\partial t} \tag{2.9}$$

Since the acceleration is known, then the total external forces on the particle can be calculated, a blob of fluid moves according to:

$$F = m \times \frac{\partial u}{\partial t} \tag{2.10}$$

The total external forces are divided into three parts: gravity *g*, pressure from other particles or vessel boundaries, and viscosity from other particles or vessel boundaries. The three external forces describe the properties of a fluid flowing through a cylindrical pipe, like flowing through a vessel in the body. The gravity is coming from Earth attraction and affects everything. Here, the pressure is assimilated to the net pressure, since the pressures from other particles and vessel boundaries may be from various directions. If the pressures cannot be balanced, or in other words, the sum is not zero, then the fluid can be pushed from the high pressure area towards the low pressure area. The pressure drop can be describes as: ΔP, and the gradient ∇P of pressure in a 3D space (∂x, ∂y, ∂z) can be describes as:

$$\nabla P = \frac{\partial \Delta P}{\partial \mathrm{x}} + \frac{\partial \Delta P}{\partial \mathrm{y}} + \frac{\partial \Delta P}{\partial \mathrm{z}} \tag{2.11}$$

Then, the gradient of pressure should be integrated over the volume of the blob to get the pressure force. Because of the assumption of incompressibility, pressure force can be considered as anything around the particle which keeps it in the constant volume V, thus, the pressure can be considered as negative and written as:

$$\mathrm{Fpressure} = -\mathrm{V} \times \left(\frac{\partial \Delta P}{\partial \mathrm{x}} + \frac{\partial \Delta P}{\partial \mathrm{y}} + \frac{\partial \Delta P}{\partial \mathrm{z}}\right) = -\left(\partial \mathrm{y}\, \partial \mathrm{z}\, \partial \Delta P + \partial \mathrm{x}\, \partial \mathrm{z}\, \partial \Delta P + \partial \mathrm{x}\, \partial \mathrm{y}\, \partial \Delta P\right) \tag{2.12}$$

The other fluid external force comes from viscosity, which can be regarded as the force trying to minimize the differences of velocity between the particle (affected by viscosity) and the nearby particles. In other words, the viscosity adjusts the velocity of the particle to the average of the nearby bits. Besides, the Laplacian $\nabla \cdot \nabla$ can measure how far a quantity is from the average values around, and since the comparable quantity is velocity, thus $\nabla \cdot \nabla u$ can be used to show the dynamic velocity rate, while ∇u shows the change of the velocity as acceleration:

$$\nabla \cdot \nabla u = \left(\frac{\partial^2 \Delta \mathrm{P}}{\partial^2 \mathrm{x}} + \frac{\partial^2 \Delta \mathrm{P}}{\partial^2 \mathrm{y}} + \frac{\partial^2 \Delta \mathrm{P}}{\partial^2 \mathrm{z}}\right) \tag{2.13}$$

Then, this should be integrated into the volume of the blob to estimate the viscosity force. Besides, a dynamic viscosity coefficient μ [25] is needed to measure viscosity of a certain kind of fluid:

$$\mathrm{Fviscosity} = -\mathrm{V} \times \mu \times \nabla \cdot \nabla u \tag{2.14}$$

Thus, equation 2.10 can be combined with equations 2.11, 2.12, 2.13 and 2.14 leads to:

$$\mathrm{F} = m \frac{\partial u}{\partial t} = mg - \mathrm{V} \nabla P + \mathrm{V} \mu \nabla \cdot \nabla u \tag{2.15}$$

Using the method of infinite difference, the mass and volume of particles go to zero. Thus, a concept of density ρ can be used to replace m and V, using the equation:

$$\rho = \frac{m}{V} \tag{2.16}$$

Inserting equation 2.16 into 2.15:

$$\rho \frac{\partial u}{\partial t} = \rho g - \nabla P + \mu \nabla \cdot \nabla u \tag{2.17}$$

Equation 2.17 is the conservation of momentum as used by particle-based simulation. The conservation of mass equation should also be considered.

Recalling the N-S (equation 2.1), $\rho \cdot u \cdot \nabla u$ is not included in equation 2.17. In fact, this term is not necessary while using particle-based rendering, because the number of particles is constant, and each mass of particle is also constant, thus, the mass conservation is guaranteed [25] and equation 2.17 is the simplified N-S equation which can describe the displacement of a fluid blob (particle).

The flow of fluid through a pipe of uniform (circular) cross-section is known as Hagen–Poiseuille flow [57]. The equations governing the Hagen–Poiseuille flow can be derived directly from the simplified N-S equation 2.17 in cylindrical coordinates by making the following set of assumptions:

- The flow is steady

$$\frac{\partial u}{\partial t} = 0 \tag{2.18}$$

- The radial and swirl components of the fluid velocity are zero.
- The flow is axis symmetric and fully developed.

Then, the N–S momentum equation 2.1 and the continuity equation are identically satisfied. The first momentum equation reduces to $\partial P/\partial r$ i.e., the pressure *P* is a function of the axial coordinate z only. Equation 2.18 reduces to:

$$\frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial u_z}{\partial r}\right) = \frac{1}{\mu}\frac{\partial p}{\partial z} \tag{2.19}$$

Where *r* is the radius of the cross-section of the pipeline radius, and u_z is the value of 3D speed *u* on the axial coordinate z. μ is the dynamic viscosity coefficient, and *P* is the fluid pressure. The 1D order solution is:

$$u_z = \frac{1}{4\mu}\frac{\partial p}{\partial z} r^2 + c_1 \ln r + c_2 \tag{2.20}$$

Where c_1 *and* c_2 are parameters to solve, and since u_z needs to be finite at *r* = 0, c_1 = 0. The no slip boundary condition [58] at the pipe wall requires that u_z = 0 at *r* = *R*, which yields:

$$c_2 = -\frac{1}{4\mu}\frac{\partial p}{\partial z} R^2 \tag{2.21}$$

Thus, combining equation 2.20 and equation 2.21 together, the parabolic velocity yields the equation:

$$u_{\mathrm{z}} = -\frac{1}{4\mu}\frac{\partial p}{\partial \mathrm{z}}(R^2 - r^2) \quad \textbf{(2.22)}$$

The maximum velocity occurs at the pipe centreline, that means *r* = 0;

$$u_{\mathrm{zmax}} = \frac{R^2}{4\mu}(-\frac{\partial p}{\partial \mathrm{z}}) \quad \textbf{(2.23)}$$

The average velocity can be calculated by integrating over the pipeline cross-section:

$$u_{\mathrm{zavg}} = \frac{1}{\pi R^2}\int_0^R u_{\mathrm{z}} \cdot 2\pi r dr = 0.5 u_{\mathrm{zmax}} \quad \textbf{(2.24)}$$

The Poiseuille equation relates the pressure drop $\Delta \mathrm{P}$ across a circular pipe of length *L* to the average flow velocity in the pipe u_{zavg} and other parameters. Assuming that the pressure decreases linearly across the length of the pipe, we have

$$\frac{\partial p}{\partial \mathrm{z}} = \frac{\Delta \mathrm{P}}{L} \quad \textbf{(2.25)}$$

Substituting equation 2.25 and the expression for u_{zmax} into the expression for u_{zavg}, and let $\Phi = u_{\mathrm{zavg}}$, yields to the equation:

$$\Phi = \frac{\mathrm{v}R^4}{8\mu}\frac{|\Delta P|}{L} \quad \textbf{(2.26)}$$

This equation is used for realistic contrast medium propagation simulation, and it is pre-computed by centreline reconstruction tool at each centreline points. Other parameters such as the length *L* and cross-section *R* are all being considered and discussed in chapter 3.

2.2.5 Methodology of blood Flow and contrast angiogram simulation

In this project, the focus is on angiographic fluid simulation and rendering. The data information about blood flow and contrast medium propagation can be obtained from real patient analysis outcomes, as well as scientific analysis and theories from many published papers. For example, Jou et al have a detailed description about transport of contrast mediums in contrast-enhanced magnetic resonance angiography [15]. Kathleen has a nice survey about the blood flow measurement and modelling [16]. Moreover, clinicians in the CRaIVE (Collaborators in Radiological Interventional Virtual Environments) consortium (http://www.craive.org.uk/) provide useful guidance for the realism of simulation, comments on the limitations and share ideas with developers. Some videos recorded during real clinical procedures, especially during the process of contrast medium injection, also give a great help for the computer-based simulation.

Supporting plentiful patient dataset sources is one of the features of our simulator, thus the contrast medium should provide the ability of propagation in various circumstances of vessel branches, and perform robustly when meeting different vessel cross-section changing rate, even sharp changes. Further, calculation such as fluid velocity in an accurate vessel model is time consuming, and in consequence this is a key problem for a real-time fluid simulation. Thus, the first task is to abstract common features for all kinds of vessel models, and to look for ways to maintain a real-time rendering without losing the realistic effects.

Blood flow in the vessels affects contrast medium in its attenuation of intensity and velocity. The natural fluid mixing between blood and contrast medium, after the injection, is not easy to achieve in a real-time since the interactions between the two fluids are complex. Researchers have developed a prototype of rendering blood flow using boids [60]. Although it works well in 2D, problems of frame speed still exist when translating to 3D, as well as dealing with the mixing simulation. Thus, I will need to consider the effect of the blood flow without losing the real-time characteristics as the second task.

The fluid can be modelled using different approaches. For example, particles can simulate the motion of molecules in the fluid. Under the N-S Equations, each kind of model can show the dynamic fluid changes of the fluid with the time. By using these models, different rendering methods like marching cubes, can be used to display in real-time on the screen the simulated angiographic images. Thus, the third task is to choose a satisfying model for a fast, accurate and qualified fluid simulation.

Chapter 4 describes the solution I use for the first task, which is a centreline reconstruction tool for post-processing the centreline generated by the 3D vascular model tool described in section 1.2. It is used for:

- Matching centreline and vessel model accurately.
- Computing heavy calculations before the simulation to accelerate the speed of the fluid simulation and reach real-time.
- Improving the expansibility and flexibility of such as bifurcation of vasculature and length of vessels.
- Improving the robustness for interacting with various vascular models.
- Enhancing the portability of the designed fluid module to different frameworks.

Chapter 5 provides the solution for the second task, which is a module designed for the control over the effect of blood flow. The different effective factors on contrast medium propagation are classified as functions. The features are as follows:

- Simulating the beating blood flow coming from the heart using object-oriented programming.
- Accelerating the rendering speed by only considering the blood flow effect, rather than simulating the fluid blood flow.
- Reducing the complexity of blood flow simulation without losing physics phenomenon.
- Enhancing the expansibility and flexibility of various kinds of blood flow simulation such as turbulent blood flow.

Chapter 6 describes the solution for the third task, providing a set of models controlling the propagation of the contrast medium, as well as combining the extracted vessel information and blood flow functions together. The contrast medium is modelled using smoothed-particle hydrodynamics and particle-based rendering techniques. The features are as follows:

- The advantages of particle based rendering as described in section 2.2.4.10.
- The simulation is similar to the real X-ray medical imaging by only performing contrast medium simulation.
- Building the propagation environment by importing from a light configuration file generated by the centreline reconstruction tool.
- Updating the fluid states by taking into account the dynamic information given by the beating blood flow module.

- Calculating the fluid simulation from simplified points and using pre-calculated hydrodynamics formula towards a real-time and realistic propagation.
- Enhancing the portability of designed module to different framework.

The concept design is provided in chapter 3 with both the functional design and module design, and the relations between the three tasks presented above. The technical organization of the project is also presented graphically. The programming language is C++, and developed in the environment of Microsoft Visual Studio 2005 (http://www.microsoft.com/en/us/default.aspx). Graphics rendering library VTK 5.0 (http://www.vtk.org/) is used to process, organize and render the medical data; while light user-interface builder library FLTK (http://www.fltk.org/) is used for a simple, clear, and compatible user interface. A more detailed description of these toolkits is also given in next chapter.

3. Concept design

This chapter describes the concept design of the simulator, and provides detailed design for the centreline (i.e. the line connecting the cross section centres of the vessels) reconstruction tool, blood flow module and contrast medium module. Chosen technical toolkits are also explained.

3.1 Overall design

Figure 3.1 describes the project architecture of blood flow and contrast medium propagation simulation. This architecture performs the link between this project and two projects developed by last year postgraduate students: a tool to generate 3D vascular models developed by Mr. Daniel Ryan King and a virtual catheterisation framework constructed by Mr. Rafal Blazewski. The brief descriptions of these two projects are given in section 3.1.1 and section 3.1.2. The three tasks mentioned in section 2.2.5 are separated in three parts, including building a centreline reconstruction tool, contrast medium module and blood flow module.

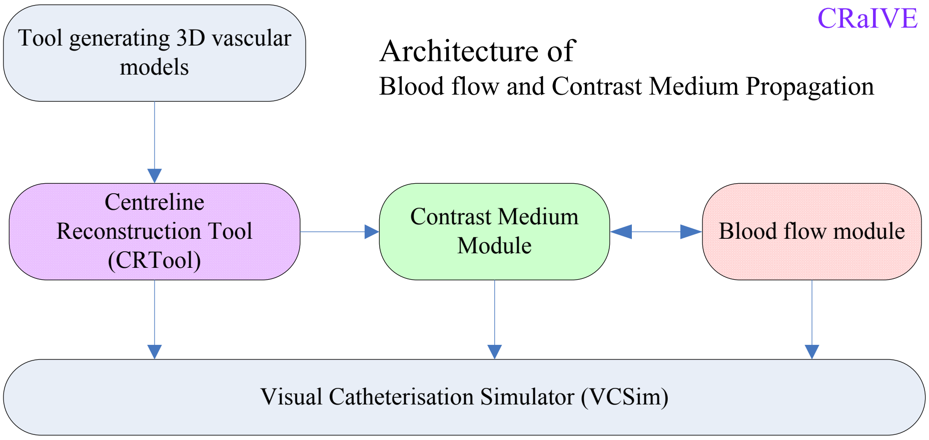

Tool generating 3D vascular models
CRaIVE
Architecture of
Blood flow and Contrast Medium Propagation
Centreline Reconstruction Tool (CRTool)
Contrast Medium Module
Blood flow module
Visual Catheterisation Simulator (VCSim)

Figure 3.1 Architecture of blood flow and contrast medium propagation simulation

3.1.1 Tool generating 3D vascular models

This project [40] provides a set of tools for generating 3D vascular models from patient datasets for interventional radiology simulations. The structure of the program provides a visual interactive method for image processing, segmentation and 3D construction. The tool also provides functions to build skeleton points and centreline of the vasculature models using a minimum spanning tree algorithm. The centreline generated is the input file for the reconstruction tool I developed as a part of this project.

3.1.2 Virtual catheterisation framework

This project [54] simulates the catheterisation procedure in interventional radiology, and provides the basic framework for my project. It models instruments as a set of particles linked by rigid springs, forming a mass-spring model, and uses new ways for collision detection and response. My project aim is to inject fluid in the loaded vasculature models and make use of the collision detection and response algorithms. Besides, the catheterisation simulator provides fluoroscopy views for a real clinical environment for contrast medium injection and propagation.

3.1.3 Centreline reconstruction tool

This tool builds the bridge between the third party tools, the 3D vascular modelling tool and the virtual catheterisation framework. This tool can not only produce standard configuration files for the simulator, but also reduce some calculation pressure for real-time simulation.

Data produced by third party tools are input into the centreline reconstruction tool which then provides different algorithms for various specific uses. It also generates various standard configuration files which can be read by the virtual catheterisation simulator. The tool is designed for expansibility for further development. Tool is also a foundation for fluid simulation of the virtual catheterisation simulator project.

3.1.4 Contrast medium module

This module is an expanded feature for the simulator. However, it is not an isolated class: it uses many basic abilities of the simulator, including I/O systems, graphics engine, physics engine, and data sources as well. Besides, this module is also designed for expansibility for further development, and different physical properties of contrast medium can be modelled on the foundation of the module framework.

3.1.5 Blood flow module

This module is also an expanded feature for the simulator, and it directly interacts with the contrast medium module. It uses the time control system of the simulator, models the beating heart, as well as output the blood flow speed and mixing parameters and other functions. These output information can be used during the propagation of contrast medium, and they are also important for modelling other clinical behaviour in the further development. Different physical properties of blood flow can be modelled in the beating heart, and output effective factors as the forms of various functions.

3.2 Concept design

Figure 3.2 is the conceptual-level of both the centreline reconstruction tool and the virtual catheterisation simulator. Six levels are illustrated, including users, I/O devices, supported device APIs, main loop, data sources and tool.

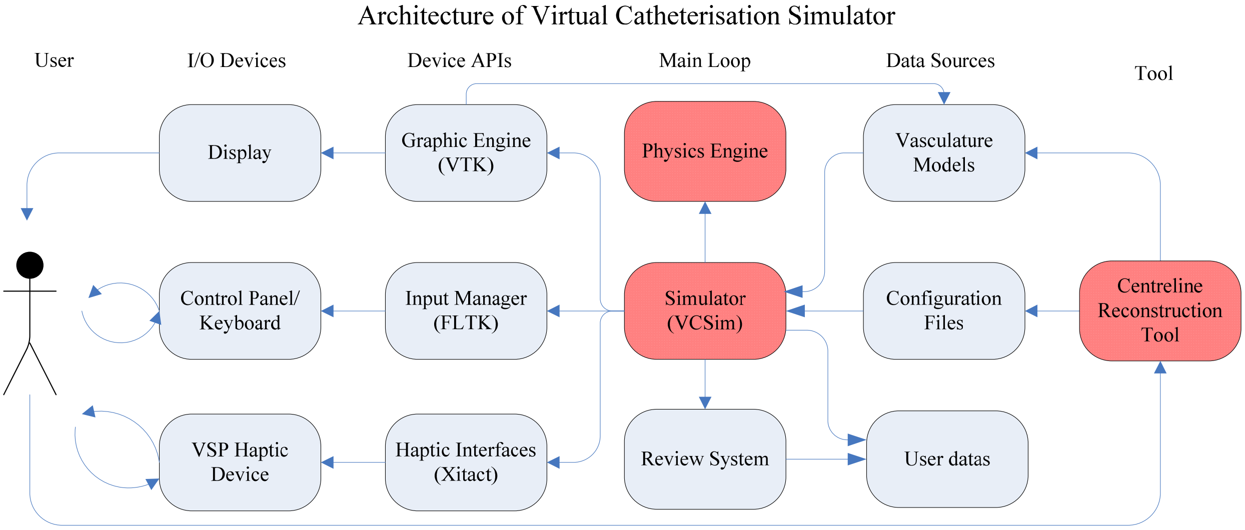


Figure 3.2 Architecture of virtual Catheterisation Simulator

3.2.1 Users

(1) Interventional radiologist

This simulator could be used by clinicians for practising the catheterization skills, and dealing with complex vascular cases in order to provide high-quality health care to patients.

(2) Students

Use this simulator as an education tutorial program offering easy access as well as the possibility to train high skill levels and various experience in term of anatomy and pathology.

(3) Reviewer

This simulator could also be used as a tool to evaluate trainee performance. Regarding the user records, it could provide certification of interventional radiologists, or it gives feedback to students during their tutorial.

3.2.2 I/O Devices

I/O devices include computer display, keyboard, control panel and VSP haptic devices, which provide interactive capability to the simulator.

The computer displays the virtual environment of the simulator, the vascular models are shown on the screen, and the users' manipulations through input devices are also shown directly on the screen. Moreover, the reviewer can see the performance of the trainees and give the evaluation.

Keyboard and control panel are the input devices the users operate directly on the computer. They allow interacting with the training programs, change different virtual environments, and control the parameters, such as blood flow speed.

The VSP haptic device, as mentioned before, is the main instrument manipulating device for users, allowing interactions with real catheter and guidewire. Users operate them just like the real clinical tools, and feel the feedback from their operations.

3.2.3 Device APIs

All the devices are supported by their own API and the framework further encapsulates them, proving an easier way for developers.

- **Graphic engine**

The graphic engine is responsible for the virtual world rendering, the geometric aspect of the vascular models, the contrast medium model, and the instruments rendering. Different graphic engines are supported by various companies and libraries. Our framework uses a library called visualization tool kit (VTK) as the rendering API.

Our framework has a graphic engine module, CVTK, which establishes the rendering pipeline, integrates the VTK rendering algorithms, and has an interface with input manager. Further, the framework provides control systems for changing the environment by adding and deleting VTK Actors, such as the vasculature or the bones. In my project, the contrast medium propagation is rendered in real-time by the graphic engine, and performs realistic effects on the computer display.

- **Input manager**

The input manager deals with the keyboard interactions. For example, it is a simple way for a user to control the simulator using some default parameters. The framework uses keyboard callback to deal with these events.

The input manager is also responsible for the graphic user interface, such as menus, buttons and scroll bars. Moreover, the capability of reaction when changing the window size, and updating the rendering area, is achieved by the input manager. The framework uses the class CFLTK to control all these abilities, and to create specific call back functions to deal with corresponding interface input. Then, thanks to the graphic engine class CVTK, the user-input effect will be directly shown on the screen.

- **Haptic interface**

The haptic interface provides the communication between user and computer with the VSP device. It includes recording the states of the device, initializing and closing the device, as well as exchanging times between the computer and the haptic device. In this project, we use the VSP haptic device and its low level library provided by Xitact (http://www.mentice.com/), encapsulated within the CVSP singleton class in the framework. It stores the main state of the device in its attributes and provides options such as VSP initialization, update of device state, applying a friction force or disconnecting the device [54].

3.2.4 Main loop

This is the core of simulation as any physical and realistic effects are achieved through this level, which loop at a given time step to evaluate the state of the simulation and reacts accordingly. In a narrow sense, the simulator can be identified as this level.

- **Physics engine**

The physical engine is organized to solve the basic physical problems in the simulation. It combines basic mathematical calculation libraries, several essential physical entities, and further achieves the advanced features such as collision detection using the mathematical algorithms and characteristics of essential entities. Figure 3.3 describes the architecture of the physical engine provided by the simulator framework.

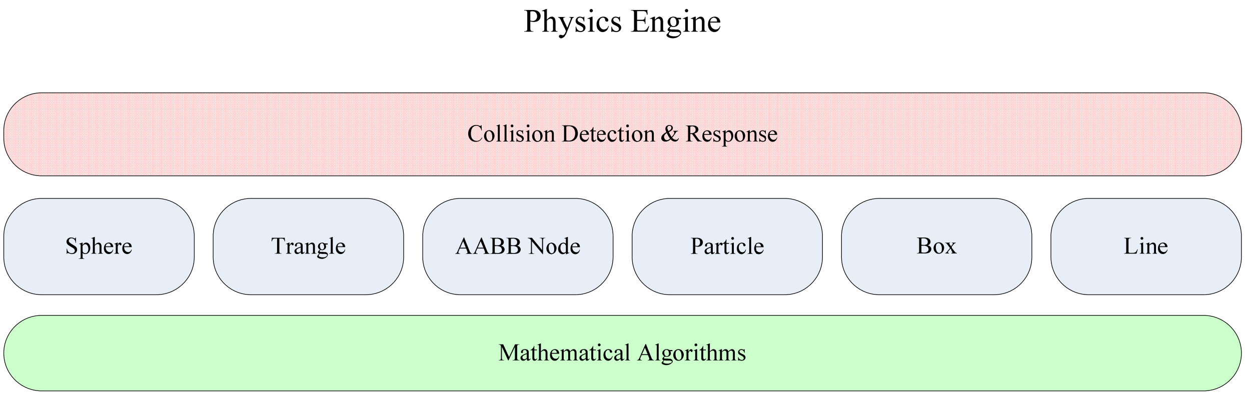


Figure 3.3 Architecture of Physics engine

(1) Mathematical algorithms

Several algorithms are integrated into the physics engine: cross-product, dot-product and distance calculation, as well as rotation, translation and so forth. They can achieve higher algorithms, for example, translation of instrument. Further, they can also be used to calculate the features for the essential entities, such as normal of plane. They are the basis of the physics engine, and can be expanded with other essential algorithms, greatly enhancing the efficiency of development. In the framework, class CVec3f is responsible for integrating mathematical algorithms.

(2) Basic graphics

The basic graphics are essential parts for advanced models, just like the bricks of the wall. For example, the contrast medium is modelled by stringing many particles, and the bifurcation is modelled with four linked spheres. Moreover, these graphics also have special features for various uses. For example, the sphere has centre point and radius, which are the basic inputs for advanced algorithms like feature detection and collision detection.

(3) Collision detection and response

The collision detection is responsible for determining the colliding parts of objects and vascular model. It includes both the broad phase collision detection and narrow phase collision detection. The first one involves the intersection check between the objects with the boxes encapsulating the objects, while the later one presents the intersection check between the objects with the triangles. The collision response represents the forces pushed on the objects which collide with vascular model using Euler or Range Kutta law depending on the required accuracy and speed. The basic idea is to calculate the new position of the objects.

- **Core Simulator**

Loop controller, schedule controller and various models form the core simulator (Figure 3.4). The loop controller is responsible for keeping the latest states of all physical models, listening to the input from users, as well as communicating with devices and graphic engines for continuous information feedback. The schedule controller builds a channel for communication between core simulator and the outside devices, input manager and graphic engines, as well as for collaboration between various models inside the core simulator. The models are divided into two kinds, one is the model with simple geometries directly modelled by basic elements from physical engine, and thus inheriting the physical features of these elements. Guidewire and catheter were already modelled, while contrast medium and blood flow are modelled and described in this report. Vessel deformation is modelled in another project by group collaborator Mr. Xiaoyuan Zhou. The other kind of model is complex and obtained from input files, such as vasculature and centreline, which are decomposed into basic elements, and thus can be checked with some simple models. Section 3.3 provides further description of the simulator.

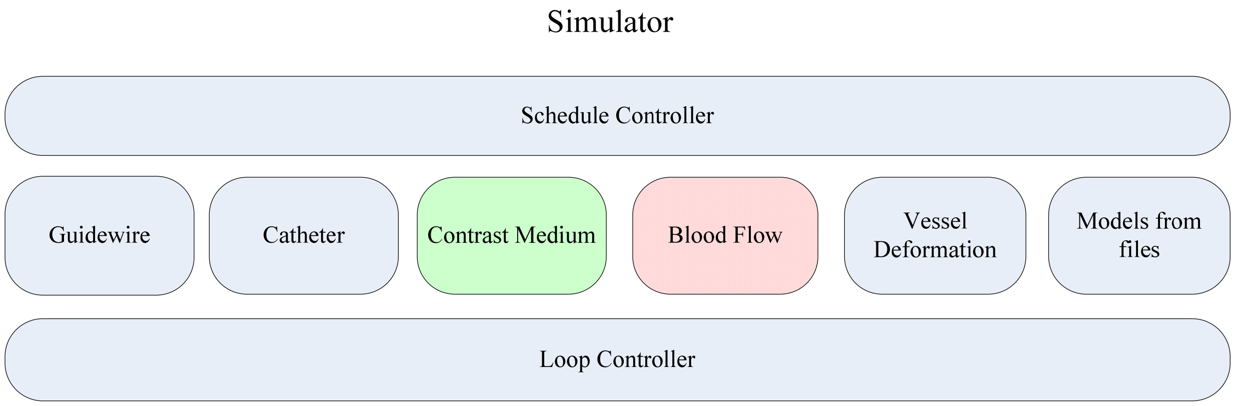


Figure 3.4 Core simulator

- **Reviewer system**

The system is left for further development. It would be used for giving marks, training recurrence and other functions needed for evaluation. Some simple features like recording historic performance of the diffusion is developed in the contrast medium module, but it will be integrated into the reviewer system later.

3.2.5 Data Sources

The data sources include the dataset files from real patients, configuration files, and the user data recording the training process. They are output and input sources for the core simulator, creating the virtual world, enhancing the realistic simulation, and improving the evaluation and tracking system. The file format includes vtk files, hdr (and img) files and txt files. Besides, designers can create new format files for specific use.

- **Vascular models**

The vascular datasets are obtained using segmentation techniques from CT volumes of real patients, which presents various pathologies, and using marching cube algorithm to create the surface of the segmented vessel datasets. Smoothing and decimating and other techniques are added for a realistic vascular model. This model normally includes thousands of triangles, depending on the precision demand. These triangles can be stored in the vascular class CModel in the framework as mentioned before, for the use of collision detection and response. The models include 3D models, fluoroscopy models, road maps, etc.

- **Configuration files**

(1) The vascular centreline

This file is an important supplement for the vascular model, as it describes the basic construction of the vessels, and can also give the curvature, cross-section, and length of the vessels. The file includes all the centreline points, their position, radius and maximum blood flow speed. Besides, the file provides the branch information, and each branch is organized as a line in the file, beginning from bifurcation points to another bifurcation, or an end point (boundary of vasculature model).

(2) Centreline configuration

This file provides information for contrast medium simulation. It includes bifurcation points, selected control points from all centreline points, and the three branches at the bifurcation points. Different configuration can also be added basing on the centreline of vasculature for various use, or regarding to other models.

(3) User data

This file is left for further development, and need connection with database to store large amount of data effectively and safely. The user data is important for evaluation, and the reviewers can record the practiced procedures of the trainees by transferring her/his data from a database through the review system.

3.2.6 Tools

Various tools are used in this simulator, providing models, centrelines and so forth. The centreline reconstruction tool is developed mainly for contrast medium propagation. Its result is communicated to the simulator using the centreline configuration file as mentioned before. This file is generated from the centreline reconstruction tool, which is specially developed for this project.

Figure 3.5 is the conceptual-level of the centreline reconstruction tool. Five levels are illustrated, including users, I/O devices, supported device APIs, main loop and data sources. The I/O devices, device APIs, and data sources are similar to the ones in the simulator. The differences are in the users and main loop. Thus these two parts are described in the next two sections.

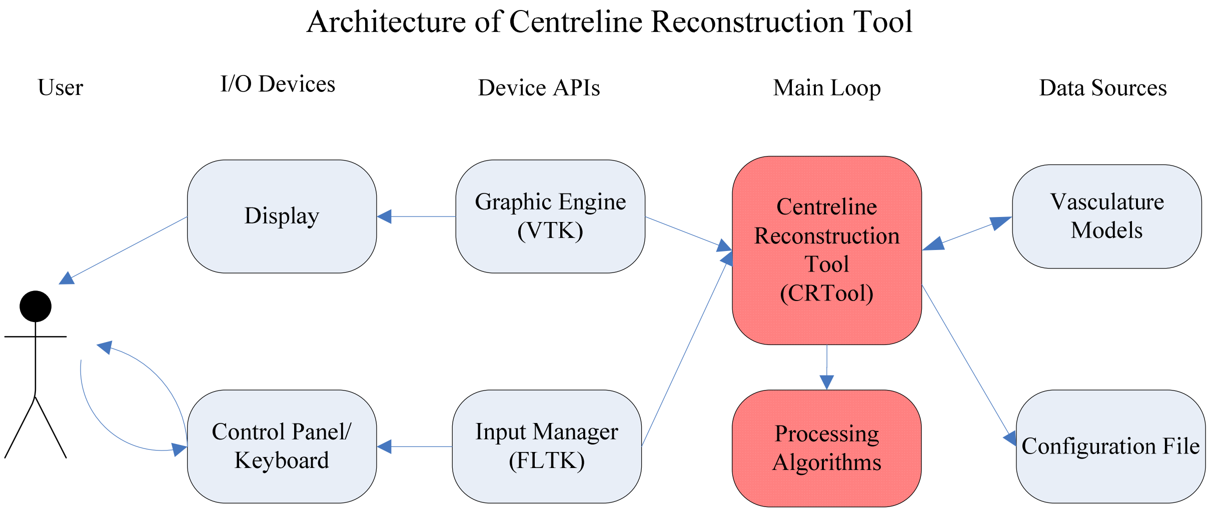


Figure 3.5 Architecture of centreline reconstruction tool

3.2.6.1 Processing algorithms

The algorithms are designed as the foundation for achieving advanced processing effects for users. One algorithm, for instance, deals with finding centreline control points and delete unnecessary control points. Some algorithms are working together. For example, a first one can show the bifurcation points and end points by using bifurcation and end points localization algorithms, while a second algorithm can then merge the centreline to connect the lines in one branch. The architecture of the processing algorithms is illustrated in figure 3.6. The implementation is described in chapter 4.

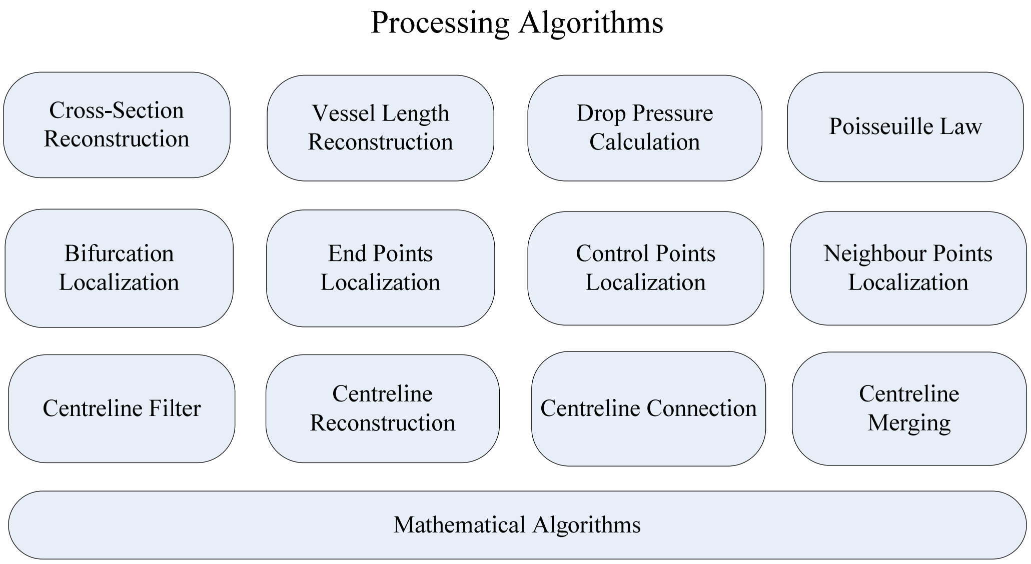

Processing Algorithms
Cross-Section Reconstruction
Vessel Length Reconstruction
Drop Pressure Calculation
Poisseuille Law
Bifurcation Localization
End Points Localization
Control Points Localization
Neighbour Points Localization
Centreline Filter
Centreline Reconstruction
Centreline Connection
Centreline Merging
Mathematical Algorithms

Figure 3.6 Architecture of the processing algorithms

3.2.6.2 Core CRTool

The core of the centreline reconstruction tool (CRTool) (Figure 3.7) is developed for loading the vascular models, providing the processing algorithm to generate and transform the centreline, and showing the graphical display. There are four kinds of transformation that the CRTool can achieve, including model transformation and rotation, centreline and cross-section reconstruction, bifurcation points and end points localization, and fluid calculation. The graphic rendering is directly updated, and the configuration files are generated or modified through the update control. The schedule module is designed as a bridge between each main parts of CRTool processing abilities, and will be described later in more detail.

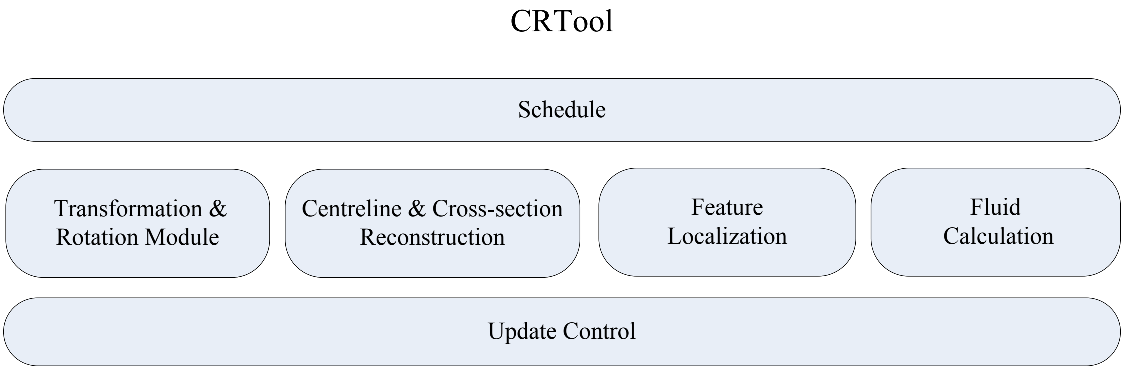

CRTool
Schedule
Transformation & Rotation Module
Centreline & Cross-section Reconstruction
Feature Localization
Fluid Calculation
Update Control

Figure 3.7 Architecture of core CRTool

3.3 Core Simulator design

There are three parts combining the core simulator system, including loop controller, schedule controller, and the different simulated models.

3.3.1 Loop controller

The loop controller (Figure 3.8) is running throughout the life time of the simulator. It keeps listening to the input manager, and transfers each update function of each model. Then it interacts with the graphic engine capability for display, as well as communicates with the VSP device interface to give feedback to the user.

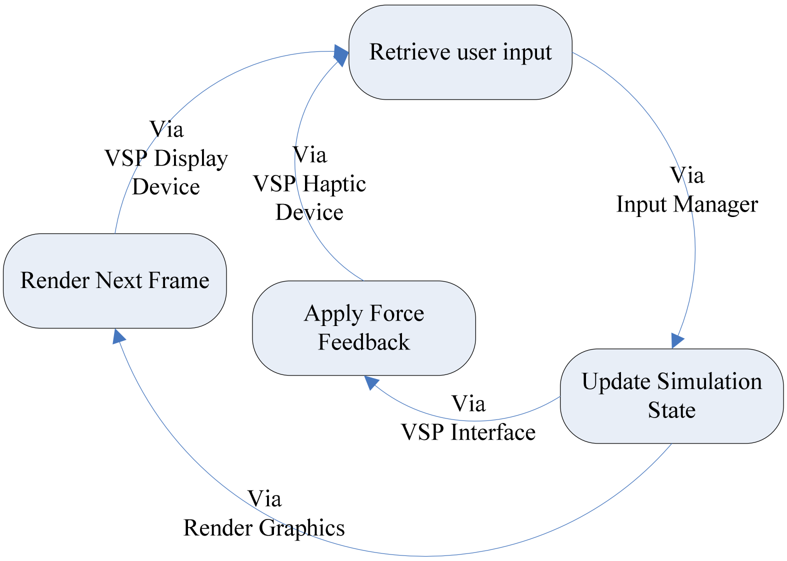


Figure 3.8 Loop controller

3.3.3 Schedule controller

The schedule controller (Figure 3.9) provides the platform or communication between the models, graphic engine, input manager and VSP haptic interface. It allows the loop controller to communicate between models and device APIs on the standard platform, and also provides the channels for communication between graphic engine, input manager and haptic interfaces. Moreover, models can be interfaced between each other through the schedule controller, to achieve the advanced effects of medical simulation. It is, for example, the case with the interactions between the blood flow and contrast medium propagation.

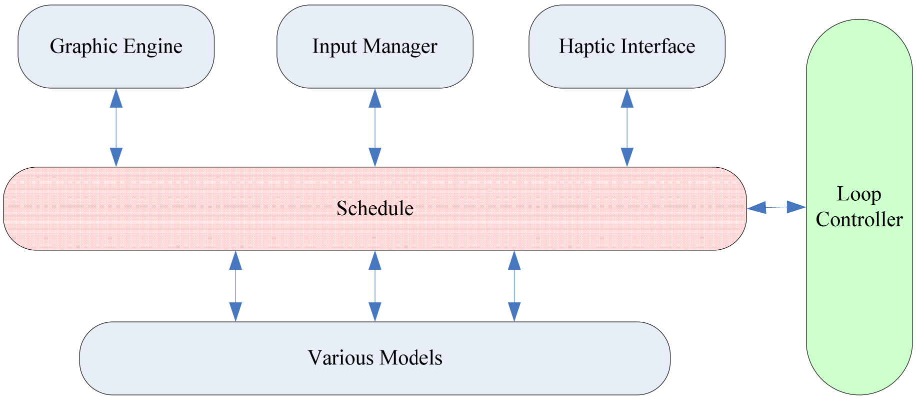


Figure 3.9 Schedule controller

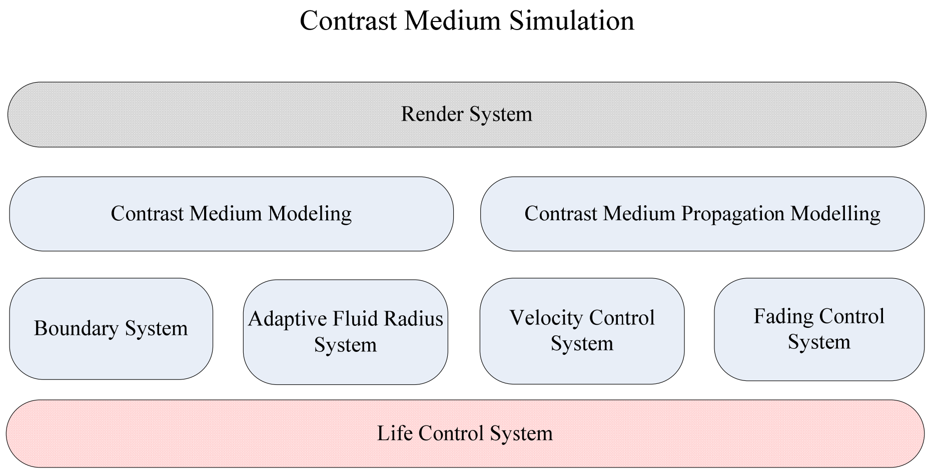
3.3.3 Models

The models are established using physical-based simulation included in the physical engine. The internal functions are designed for geometrical construction, physical properties, as well as rendering and dynamic update. The guidewire, catheter, vasculature modules are modelled and well documented in [54 62], while the centreline, contrast medium and blood flow modules are described in the following sections.

3.3.3.1 Contrast medium

The contrast medium simulation includes the simulation of fluid shapes, fluid propagation and attenuation. In order to achieve these factors, various modules are designed for different factors of fluids.

Figure 3.10 describes the contrast medium simulation. The contrast medium modelling module simulates the basic shape of the contrast medium, while the contrast medium propagation modelling module simulates the behaviour of the contrast medium. Four physical-based modules are designed for changing factors for both the basic shape, and the basic behaviour of the contrast medium. The boundary system is responsible for keeping the fluid inside the vasculature, with an adaptive fluid radius system controlling the shape of the contrast medium. A velocity control system is used for calculating the current contrast medium propagation speed, while a fading control system achieves the effect of attenuation during the propagation. Further, regarding to the four physical-based modules, the life control system updates and tracks the shape and behaviour of contrast medium, and provides the frame time as an important parameter participating in physical calculation. The rendering system draws the shape and behaviour of contrast medium, as well as shows the attenuation effect using opacity and colour. The implementation is described in chapter 5.


Contrast Medium Simulation

Render System

Contrast Medium Modeling

Contrast Medium Propagation Modelling

Boundary System

Adaptive Fluid Radius System

Velocity Control System

Fading Control System

Life Control System

Figure 3.10 Architecture of contrast medium simulation

3.3.3.2 Blood flow

Figure 3.11 describes the blood flow model simulation. The velocity of blood flow is variable in this project, to model beating hearts of different real patients, simulating different heart rates. Since the heart rate is an important factor for blood flow speed and the physical phenomenon of mixing with contrast medium, it is isolated to provide the basic parameters for another two modules, blood flow velocity control system and mixing system. They are designed for simulating the blood flow speed at each time frame under different beating heart models, and simulating the mixing effect with the contrast medium model. The life control system is similar to that of contrast medium, it is inspired from the beating heart from a real person, keeping the heart continuously working and controlling the behaviour of other models which regards blood flow as an impact factor during their simulation.

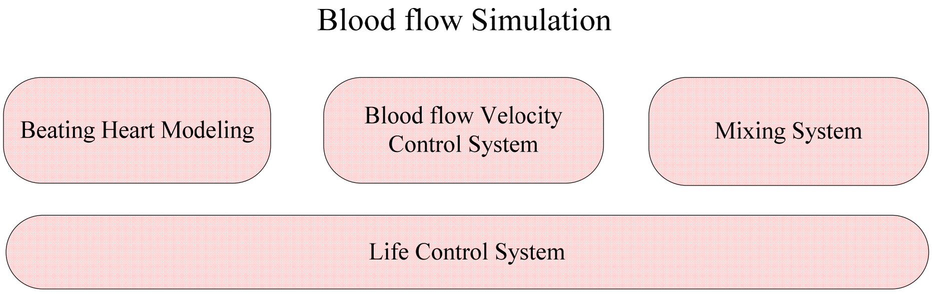


Figure 3.11 Architecture of blood flow simulation

3.3.3.3 Centreline model

The centreline model (Figure 3.12) is similar to the vasculature model class, CModel, already developed in the framework. Centreline information package uses data structures, such as arrays and linked lists to store the information read from centreline file and configuration file. The skeleton points are stored in a structure including radius, x, y, z and maximum blood flow speed, while the centreline branches and feature branches are stored in a 2D linked list. Feature points, and control points are stored in two 1D linked lists. Centreline modelling uses the structure information for building the vascular skeleton using basic line elements coming from the physical engine, and can be rendered through the rendering system. Regarding the outside information inputted in the centreline model, various modules are designed for different interaction simulation, and the detailed implementation will be discussed in chapter 5.

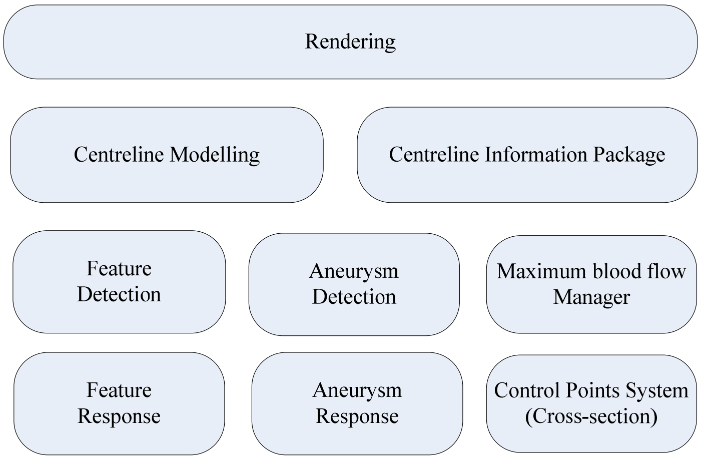


Figure 3.12 Architecture of centreline model

The collision detection and response were so far confined to the physical engine but, in this project, the feature detection and response is also used in the fluid simulation, thus, these two modules are now also designed in the centreline model. They are responsible for dealing with the interactions when contrast medium meets the bifurcations. Similarly, for an aneurysm, detection and response modules are designed for dealing with the disease when the size of vasculature cross-section has a very large gradient change. The maximum blood flow manager is responsible for transferring the maximum blood flow value to the contrast medium module regarding to the position information of contrast medium fluid. The control points system provides the capability for checking the cross-section radius where the contrast medium particles lie with respect to the control points selected from the centreline points.

3.3.3.4 Interaction between models

The simulation of contrast medium propagating in the blood flow is achieved as collaboration between not only the models I designed, but also the existing models in the simulator framework. Figure 3.13 illustrates the interface design for interaction between models. There are five models involved, and the direction of arrows describes the orientation of parameter transfer. The names of interface products are shown under the diagram regarding to the capital letters.

- **Catheter with Contrast medium**

One interaction is designed between these two models, and there is transfer when the contrast medium is just injected from the catheter. The position of the first two control points of the catheter is known and the initial position of contrast medium is thus initialized.

- **Vessel centreline with contrast medium**

Six interactions are designed between these two models. In the initialization, initial position of contrast medium is transferred to vessel centreline model, and the initial radius of particle is obtained, which forms the first particle of contrast medium. Then at each time frame, the contrast medium model transfers the first particle to the vessel centreline to check if there is a collision with a vascular bifurcation. And if it is the case, then the response information will be transferred to the contrast medium with the feature location, as well as the three feature branches. Further, contrast medium can visit all the basic centreline information, including the position, radius, and max blood flow speed.

- **Blood flow with contrast medium**

Three interactions are designed between these two models. At every time frame, max blood flow speed is transferred to the blood flow module and current actual blood flow velocity and the mixing parameter are obtained immediately after the computing of the internal system.

- **Vascular model with contrast medium**

Thanks to the interfaces designed in the framework, at every time frame, the first particle is transferred to the vascular model for collision checking. If there is a collision, the vascular model returns the colliding triangles, as well as the distance between the first particle and the surface.

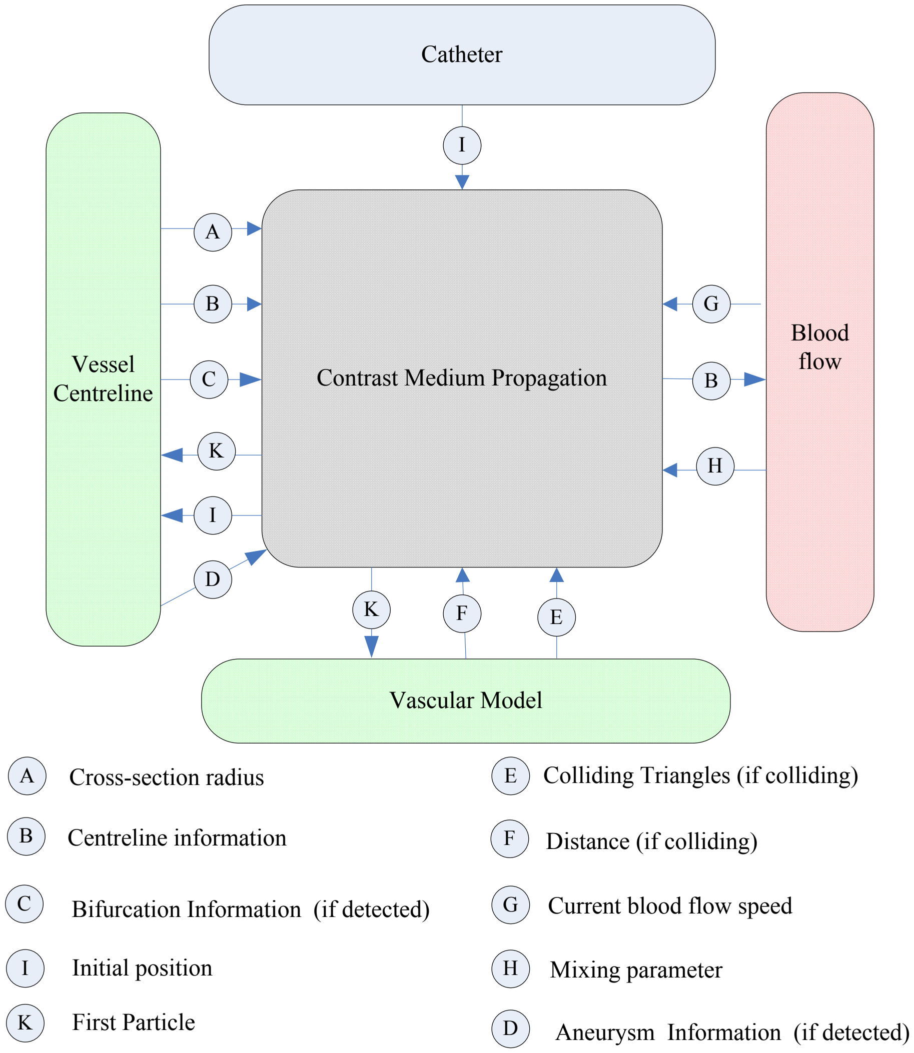

Catheter
I
Vessel
Centreline
A
B
C
K
I
D
Contrast Medium Propagation
G
B
H
Blood
flow
K
F
E
Vascular Model
A Cross-section radius
B Centreline information
C Bifurcation Information (if detected)
I Initial position
K First Particle
E Colliding Triangles (if colliding)
F Distance (if colliding)
G Current blood flow speed
H Mixing parameter
D Aneurysm Information (if detected)

Figure 3.13 Interface design for interaction between models

3.4 Development technologies

In this part, the main toolkits and libraries are introduced. Microsoft Visual Studio 2005 is the development environment for our project. During this project, FLTK and VTK libraries are included and linked to access their functions and classes. FLTK is used for building the user-interface (UI), while VTK is a visualization ToolKit mainly for image processing and rendering. Besides, CMake is also used for generating FLTK and FTK libraries and link them in our project, with Microsoft Visual Studio 2005.

3.4.1 Microsoft Visual Studio 2005

This widely used framework has useful and powerful functions and features. It supports software design and implementation, and also sustains software tests and management. Besides, it provides the possibility of using different development languages, including C++, C#, VB and so forth. It also facilitates team project with a good code management and software configuration management. From a C++ programmer's viewpoint, it is simple to see the architecture of the whole classes of the projects as well as its functions, and it has an excellent debugging system which makes it easy to correct and improve the code. Moreover, different libraries can be included and linked in this framework, developers can use its basic functions to add different features and achieve various effects on their workspaces.

3.4.2 CMAKE

Cross-Platform Make (CMake) (http://www.cmake.org/) is a cross-platform, open source build system [64]. It allows the generation of native build files, such as makefiles for Unix systems and project workspaces for Microsoft Visual Studio. CMake has the capability to look for the include sources needed to be linked for the build files, or users can choose the include files or output locations by a simple user interface to control the building process. Moreover, it can also test the chosen compiler for supported features that may be necessary for a successful build. In our project, CMake successfully build the FLTK and VTK libraries from source files to native build files of workspaces for Microsoft Visual Studio 2005. All libraries were included and linked with the framework automatically.

3.4.3 FLTK

Fast Light Toolkit (FLTK) (http://www.fltk.org), provides modern GUI functionality with native OpenGL rendering support and built-in GLUT emulation [63]. It is a cross-platform C++ GUI Toolkit for Linux/Unix, Microsoft windows, and Mac OS system. FLTK is small and modular enough to be easily linked. FLTK also provides a nice UI builder which makes the user-interface development much easier. Our project rendering window is set under the FLTK 1.1.7 interface. This open-source software shows our different kinds of rendering modes as a set of clear menu buttons and static working states.

3.4.4 VTK

The visualization Tool Kit (VTK) (http://www.vtk.org) provides a wide variety of visualization algorithms for scalar, vector, tensor, texture, and volumetric data [65]. Following the VTK's visualization pipeline, it includes data reading, data processing, image segmentation, contouring, polygon reduction, mesh smoothing and decimation, Delaunay triangulation, and rendering. Different advanced algorithms and modelling techniques are integrated in the libraries, as well as basic data representations, such as vertex, voxel, line, triangle and so forth. In our project, VTK 5.0 is used to render blood flow and contrast agent propagation.

4. Centreline reconstruction tool

The vascular centreline provides prior information for contrast medium propagation. A developed centreline reconstruction tool is described in this chapter. It provides two standard files to the fluid simulation framework. The centreline file includes vascular branches, and skeleton points with radius, position and maximum velocity, while the centreline configuration file information includes bifurcation feature points, bifurcation branches, and control points.

4.1 Centreline transformation and rotation

The centreline can be considered as the skeleton of the vasculature. It can be generated from real patient datasets and loaded into the simulator framework. However, the coordinates of both the vascular surface model and the centreline are generated from the 3D vascular model building tool and they are not suited for the coordinate system of the simulator framework. In the initialization of the framework, the vascular model should be well localized and matched with the initial position of the instruments. Moreover, the centreline should be matched with the vascular model as well to obtain an accurate and well located vascular surface model. Thus, one of the important modules of the centreline reconstruction tool is developed for centreline transformation and rotation.

The function can be achieved by using the following equations:

Translation:

$$[x^T, y^T, z^T, 1] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ dx & dy & dz & 1 \end{bmatrix} [x, y, z, 1]$$

Rotation:

$$[x^R, y^R, z^R, 1] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\alpha & \sin\alpha & 0 \\ 0 & -\sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\beta & 0 & -\sin\beta & 0 \\ 0 & 1 & 0 & 0 \\ \sin\beta & 0 & \cos\beta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\gamma & \sin\gamma & 0 & 0 \\ -\sin\gamma & \cos\gamma & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} [x^T, y^T, z^T, 1]$$

Where (x, y, z) is the position of each centreline points, and (x^T, y^T, z^T) is the updated position after translation, and (x^R, y^R, z^R) is the updated position after rotation. The translation algorithm is organized as a 4×4 matrix, while the rotation algorithm is organized as the multiplication of three 4×4 matrices, and α, β and γ are representing the rotation angles in radian along the axis x, y and z, respectively.

4.2 Centreline and cross-section reconstruction

The module of centreline and cross-section reconstruction is developed for a well-organized centreline file without unnecessary lines, without disconnected centrelines, and without redundant centrelines. Besides, the cross-section on the centreline points should be well modified. Figure 4.1 provides the basic operation process for this aim. It includes centreline filter, centreline reconstruction and cross-section reconstruction. Each part is described in the following sections.

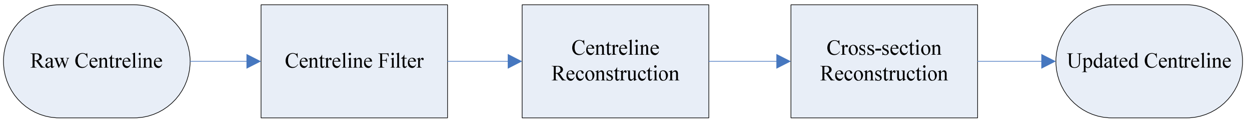
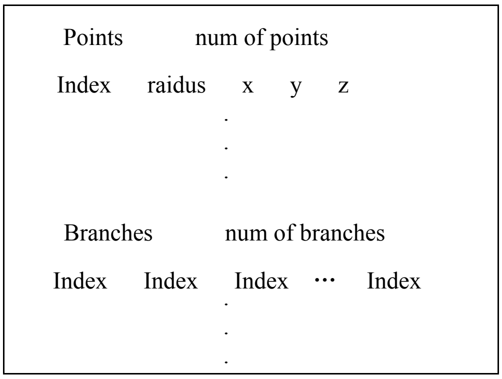


Figure 4.1 The process of centreline and cross-section reconstruction

4.2.1 Raw centreline

The raw centreline information is organized in a txt file, and its contents are described in Figure 4.2. The data under the title 'Points' is the set of centreline points, organized by index, radius, x, y, and z. The data under the title 'Branches' is the set of centreline branches, where each line represents an actual centreline line with indexes of all the centreline points in this branch. One point is stored in an object of a structure Point, which has six members, there are index, radius, x, y, z, and velocity. The values of velocity are not in the raw file, thus the initial values are all -1, but it will be further evaluated in the centreline velocity calculation phase. Moreover, all centreline points are stored orderly in a linked list. Besides, a line of branch is stored in a linked list where the member is index, while all branches are stored in a linked list as well, but where the member is branch.



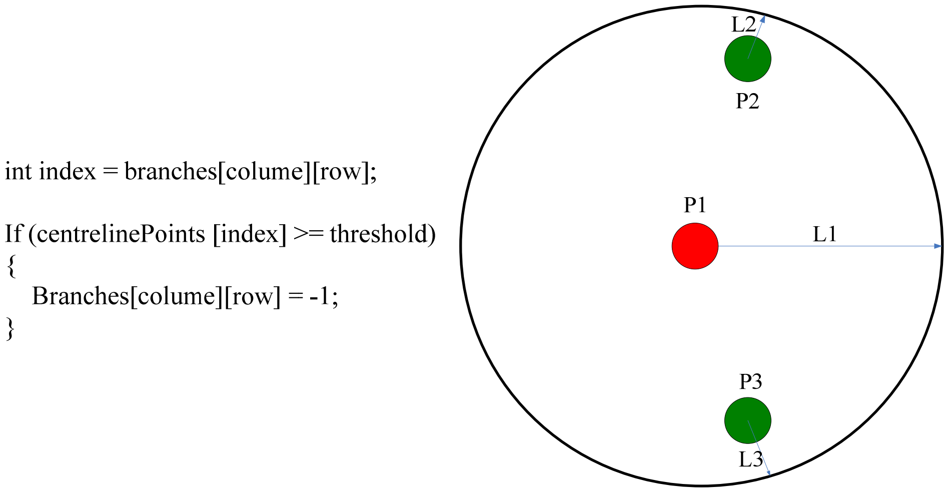
```
 Points        num of points

Index    raidus    x    y    z
                .
                .
                .

 Branches         num of branches

Index    Index    Index  ···   Index
                .
                .
                .
```

Figure 4.2 File format of raw centreline file

4.2.2 Centreline filtering

- **Radius filtering**

Figure 4.3 describes the idea for filtering some unnecessary centreline points using the cross-section radius information. P2 and P3 are deleted from the centreline points based on their cross-section radius L2 and L3, respectively. The two lengths of the cross-section radius are under the threshold value entered by the user, thus these two points are considered as non-centreline points and will not be written into the output updated centreline file. P1 is protected as a good centreline point since L1 is longer than the threshold. This algorithm runs when reading the branch information of the centreline, a lot of unnecessary points are filtered and marked as -1. The algorithm is described on the left of figure 4.3:



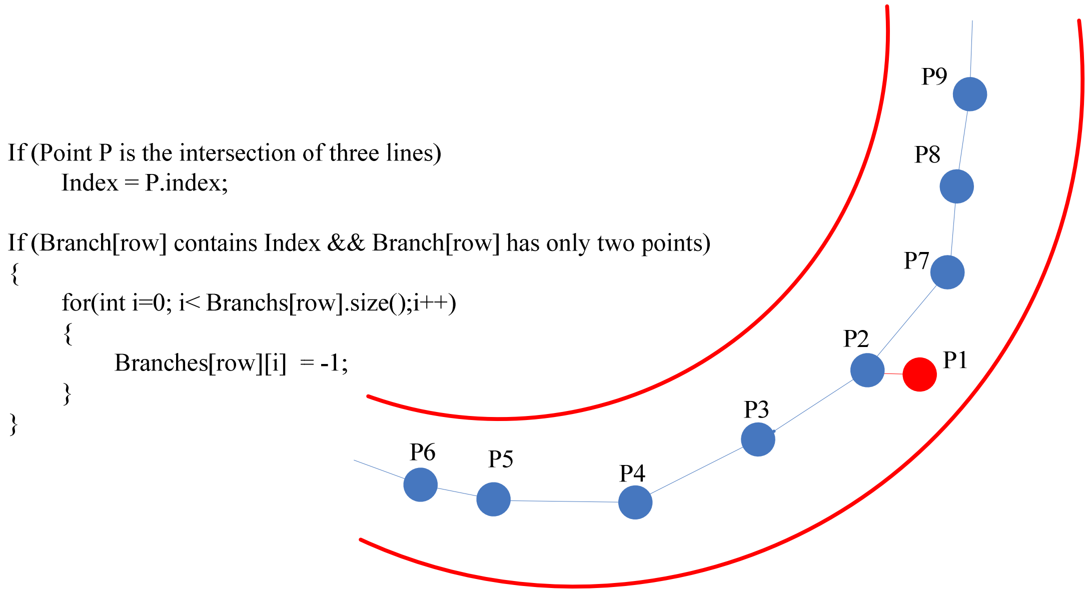
```
int index = branches[colume][row];

If (centrelinePoints [index] >= threshold)
{
   Branches[colume][row] = -1;
}
```

Figure 4.3 Radius filtering schematic diagram

This algorithm works well at the very beginning of the filtering process, which filters a large amount of unnecessary centreline points. However, some thin vessel branches may be over-filtered by using this algorithm, thus the following two algorithms are also introduced.

- **Filter unnecessary lines**

Figure 4.4 describes the idea of filtering unnecessary lines, which are isolated lines attached to some centreline points, line (P1, P2) in the diagram. In this case, P1 is not fitered by the radius filter. Another filter is thus needed for automatically processing it. P2 is the intersection points among three branches, in the diagram, there are branch_A(P2, P7, P8, P9…), branch_B(P2, P3, P4, P5, P6…) and branch_C(P2, P1). Among three branches, branch_C has only two points, thus this branch should be deleted and both two indexes of points are set to -1.



```
If (Point P is the intersection of three lines)
      Index = P.index;

If (Branch[row] contains Index && Branch[row] has only two points)
{
      for(int i=0; i< Branchs[row].size();i++)
      {
            Branches[row][i]  = -1;
      }
}
```

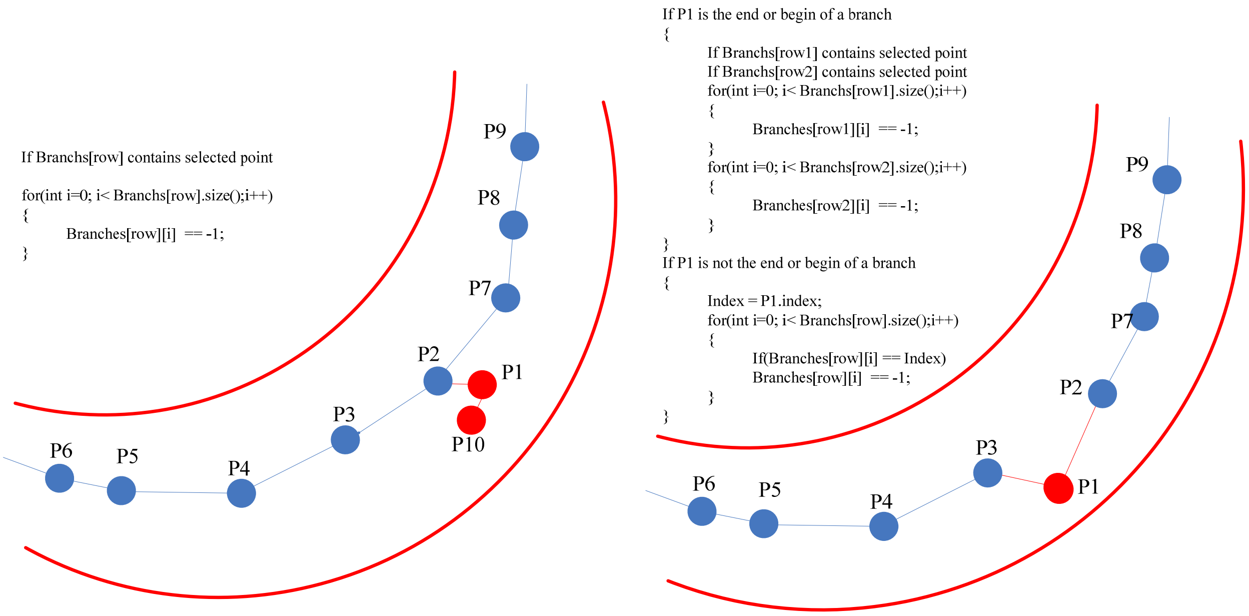
Figure 4.4 schematic diagram of filtering unnecessary lines

The advantage of using this algorithm is that after radius filter, it cleans a large amount of unnecessary lines fastly and automatically. But, in some cases, small bifurcations are ignored after using this algorithm if a short branch of vessel is organized with only two centreline points. Moreover, the algorithm cannot delete other unnecessary lines if they are organized with more than one centreline point.

Thus a third phase is introduced to delete those unnecessary lines manually.

- **Delete unnecessary points**

Figure 4.5 describes the two cases where the algorithm deleting unnecessary points manually is needed. So far, a large number of unnessary points and lines are filtered by the previous two algorithms, but the vasculature is so complex that filtering manually is necessary for some cases. As mentioned before, if two unnecessary points form an unnecessary branch, then the simple algorithms discussed above can not detect this situation. Thus, deleting P10 by hand, and then using that automatic algorithm, then Branch (P2, P1, P10) can be totally filtered. In the other case shown in Figure 4.5, P1 is an unnecessary point, and it is not the intersection between three branches, then this algorithm provides the function for deleting the point P1. The two resulting lines will be filtered and merged when saving the result file.



```
If Branchs[row] contains selected point

for(int i=0; i< Branchs[row].size();i++)
{
        Branches[row][i]  == -1;
}
```

```
If P1 is the end or begin of a branch
{
        If Branchs[row1] contains selected point
        If Branchs[row2] contains selected point
        for(int i=0; i< Branchs[row1].size();i++)
        {
                Branches[row1][i]  == -1;
        }
        for(int i=0; i< Branchs[row2].size();i++)
        {
                Branches[row2][i]  == -1;
        }
}
If P1 is not the end or begin of a branch
{
        Index = P1.index;
        for(int i=0; i< Branchs[row].size();i++)
        {
                If(Branches[row][i] == Index)
                Branches[row][i]  == -1;
        }
}
```

Figure 4.5 Manually deleting unnecessary points forms an unnecessary line P1P10 (left), or two unnecessary lines P3..P6 and P2P7..P9 (right).

4.2.3 Centreline connection and merge

After using the previous three methods for filtering unnecessary lines and points, some gaps appear between centreline points. Thus, algorithms of centreline connection and merging are designed for connecting the centreline points if there is a gap, and merging the centreline points if they belong to one branch. An automatic algorithm is developed by my collaborator Xiaoyuan Zhou [62], and is combined in the tool as the automatic connection and merge function. However, for some complex models, connecting and merging by hand is necessary.

Figure 4.6 describes the two cases of centreline connection and merging. The automatic algorithm works well for the simple cases (left diagram, figure 4.6) just like the right case of figure 4.5 after deleting point P1. Connection between P2 and P3 can be judged regarding to the similar direction between P4 and P3, with the direction between P2 and P7. However, complex cases (right diagram, figure 4.6) are not easy to judge and sometimes wrong connections occur (connect P2 with P7) since the direction between P2 and P3 is similar to direction between P2 and P7 rahter than P2 and P1, where P7 and P1 are both candidate points to be connected. Thus, manual connection algorithm should deal with these complex cases, and L1 and L2 are connected in this case rather than L3.

Merging can be completed when using the automatic connection and merging algorithms. However, if connecting the points by hand, users should select the begin and end points to merge a set of lines. The begin point is always a bifurcation point, while the end point can be an end point or a bifurcation point.

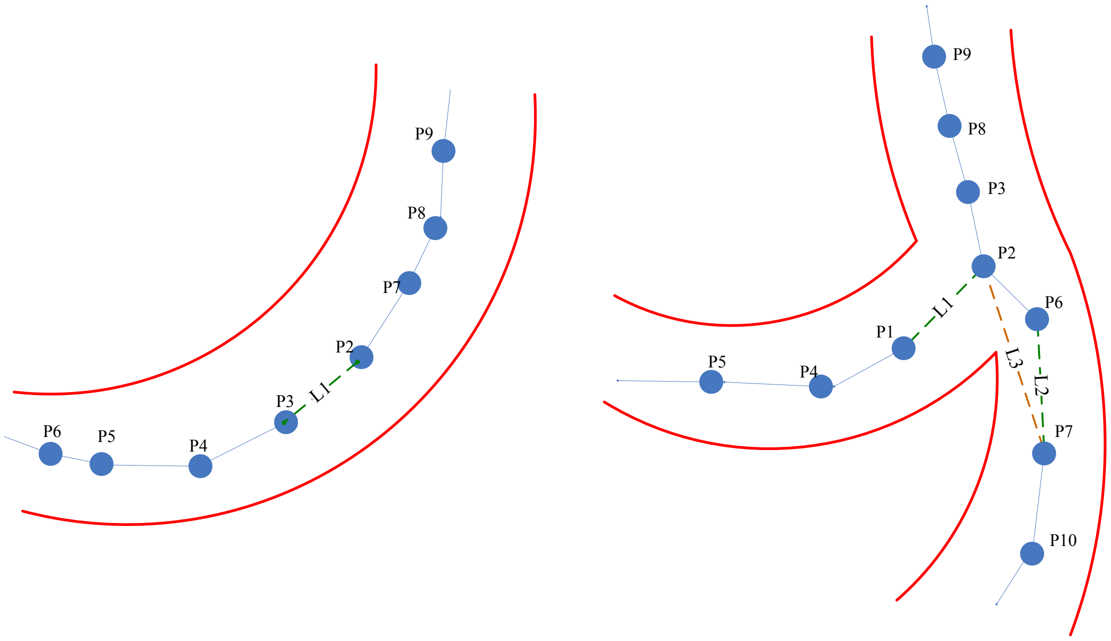


Figure 4.6 Centreline connection, automatic algorithm for connection (left), connection by hand (right)

4.2.4 Cross-section reconstruction

The contrast medium model uses the vascular cross-section radius to control the size of the fluid particle for filling the vasculature without apparent gaps between surface of fluid and internal surface of vessels. The radii are obtained from the centreline points information, and are stored in the Point structure after reading the raw centreline from the input file. However, most of the radius are larger than the real vascular cross-section radius information, thus the algorithm for cross-section reconstruction is necessary, especially in the three cases described in figure 4.7. Normal centreline points are centreline points other than bifurcation points and neighbour points around bifurcation points, which have unusually large radius that it needs to be changed to a smaller radius set by users. Bifurcation points and their neighbour points are important for contrast medium motion as described in chapter 6, thus they can be corrected one by one accurately so as to get a satisfactory radius, relevant for simulation steps.

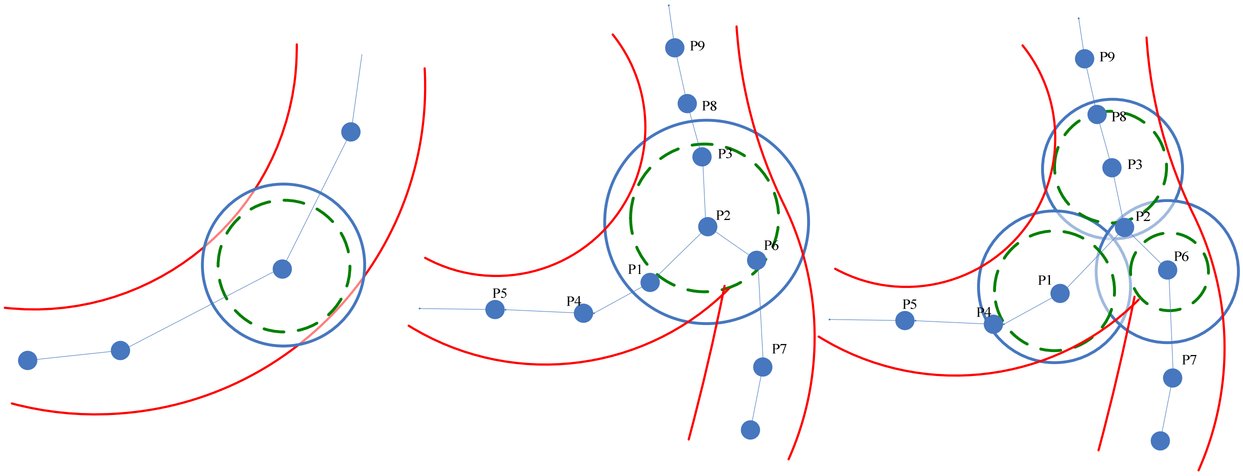


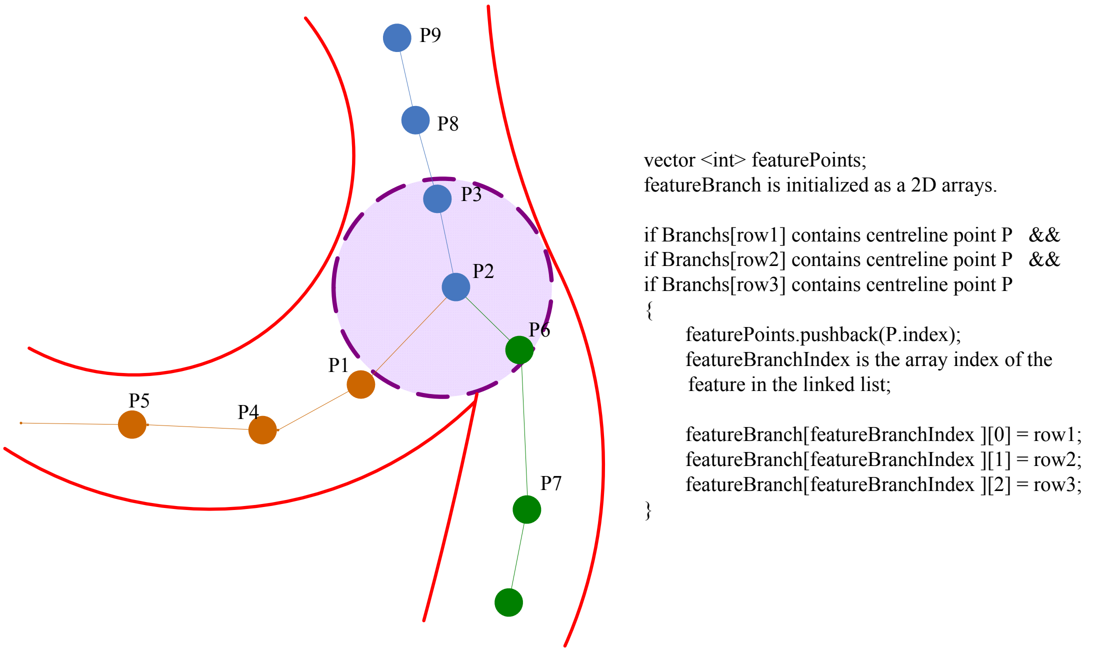
Figure 4.7 Cross-section radius reconstruction of normal centreline points (left), of bifurcation point (middle), of neighbour points around a bifurcation point (right).

4.3 Feature, end and control points localization

Feature points are bifurcation points. End points have only one intersection with the whole centreline, representing the very end of vasculature models. Control points are selected from the large amount of filtered centreline points, and their number can be specified by users at various sampling rates.

4.3.1 Feature points localization

Thanks to the algorithms described in section 4.2, a set of lines between bifurcations, and between bifurcation and end points are merged into one branch line. Thus, the feature points can be localized by searching the centreline points which are the intersection between three different branches. Figure 4.8 describes the idea and provides the algorithm. The sphere around point P2 will be shown graphically when the feature point P2 is localized by running the algorithm, and the radius of the sphere represents the cross-section radius of that point. Branch_A (P2, P3, P8, P9…), Branch_B (P2, P1, P4, P5…), and Branch_C (P2, P6, P7…) are three branches in figure 4.8. P2 is the only common point of the three branches, thus P2 is detected and stored in the featurePoints linked list. Together with other feature points being detected and localized, P2 will be written into a centreline configuration file which will be discussed in section 4.5. Besides, the three branches containing that feature point are also stored. They are important points for contrast medium simulation as will be explained in chapter 6.

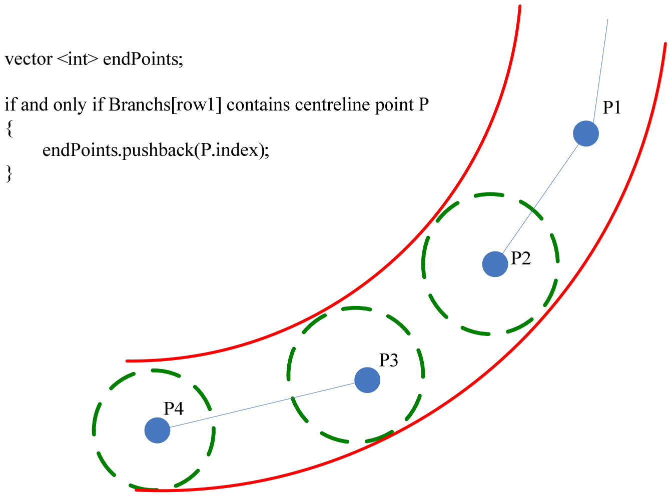


```
vector <int> featurePoints;
featureBranch is initialized as a 2D arrays.

if Branchs[row1] contains centreline point P   &&
if Branchs[row2] contains centreline point P   &&
if Branchs[row3] contains centreline point P
{
       featurePoints.pushback(P.index);
       featureBranchIndex is the array index of the
        feature in the linked list;

       featureBranch[featureBranchIndex ][0] = row1;
       featureBranch[featureBranchIndex ][1] = row2;
       featureBranch[featureBranchIndex ][2] = row3;
}
```

Figure 4.8 Feature points Localization

4.3.2 End points localization

End points are used for defining the boundary of the vascular model, and also to localize candidate points for connection and merging. Thus end points are important points for centreline connection and merging, but they are not written into the configuration file for fluid simulation. The algorithm for end points localization is described in figure 4.9, this kind of points has only one intersection with all the branches of the vasculature. Thus, P2, P3 and P4 are localized and graphically outlined with green spheres. The real end point (boundary of model) from these three points is P4 and is stored in the endPoints linked list, while P2 and P3 are pseudo end points and can be connected manually as mentioned in section 4.2.3.

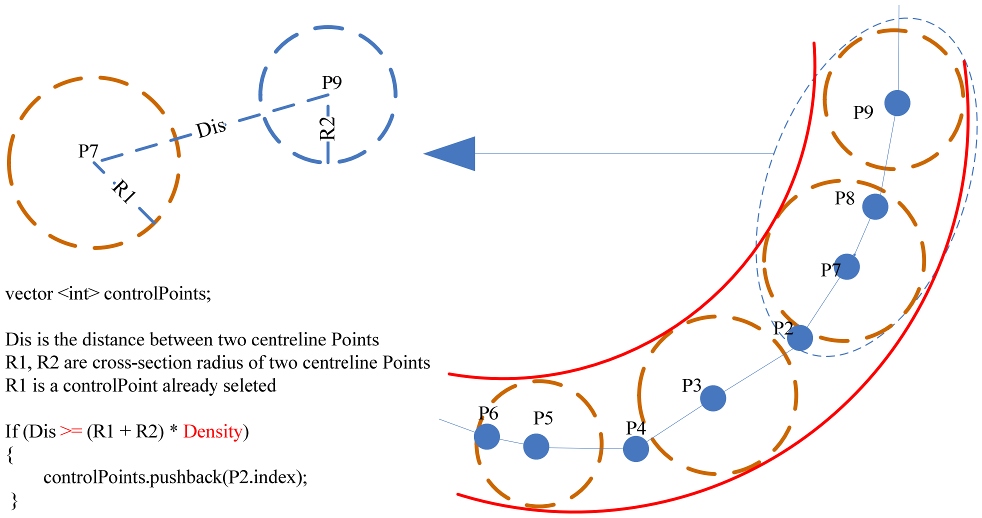


```
vector <int> endPoints;

if and only if Branchs[row1] contains centreline point P
{
      endPoints.pushback(P.index);
}
```

Figure 4.9 End points localization

4.3.3 Control points localization

Control points are used for calculating the initial fluid cross-section radius, and will be written into the centreline configuration file. They are selected from the large amount of filtered centreline points, depending on the Density or sampling rate entered by the users. The algorithm and the idea of control points localization are described in figure 4.10, where P9 is localized as a control point since the distance between P9 and P7 (alreay in the controlPoints linked list) is longer than multiplication between the sum of their individual cross-section radius and a Density parameter. By specifying this parameter, the number of control points can be controlled. For example, if the density is low enough, P8 can also be selected as a control point although it is not selected in the case of figure 4.10. The control points are localized and graphically shown with spheres, and the algorithm deleting unnecessary points, as mentioned in section 4.2.3, needs this localization step as a medium for selection and deletion.



```
vector <int> controlPoints;

Dis is the distance between two centreline Points
R1, R2 are cross-section radius of two centreline Points
R1 is a controlPoint already seleted

If (Dis >= (R1 + R2) * Density)
{
      controlPoints.pushback(P2.index);
 }
```

Figure 4.10 Control points localization

4.4 Computing velocity of centreline points using Poisseuille Law

Equation 2.3 provides the velocity expression described by using the Poisseuille Law, and in section 2.2.5 a set of deductions from the N-S equation to Poisseuille Law are explained. In my project, this equation is used for calculating the maximum blood flow speed at centreline points and will be written into centreline points as the sixth scalar value attached to the centreline points information.

By using the reconstruction algorithm, the standard format of centreline is obtained. Further, by using the point localization algorithm, different branches are partitioned. This information is important for the following processes and accurately computing velocity. Figure 4.11 describes the two key factors for computing velocity using the Poisseuille Law, length and pressure drop in the vasculature. In order to calculate the branch length between P1 and P6, and P1 and P11, two branches are involved based on the previous work. Branch_A(P1, P2, P3, P4, P5, P6) and Branch_B (P1, P7, P8, P9, P10, P11), where P1, P6 and P11 are all feature points and highlighted by three bounding spheres. The length L1 and L2 are lengths of branch_A and branch_B, and are calculated using approximation methods by summing the length between the centreline points, i.e. between two feature points. Thus, in the diagram, L1 is approximated to *Len*(P1, P2) + *Len*(P2, P3) + *Len*(P3, P4) + *Len*(P4, P5) + *Len*(P5, P6), where Len represents the length between two centreline points. Since the branch is well organized in a linked list, the calculation is simple and efficient. L2 can be calculated similarly.

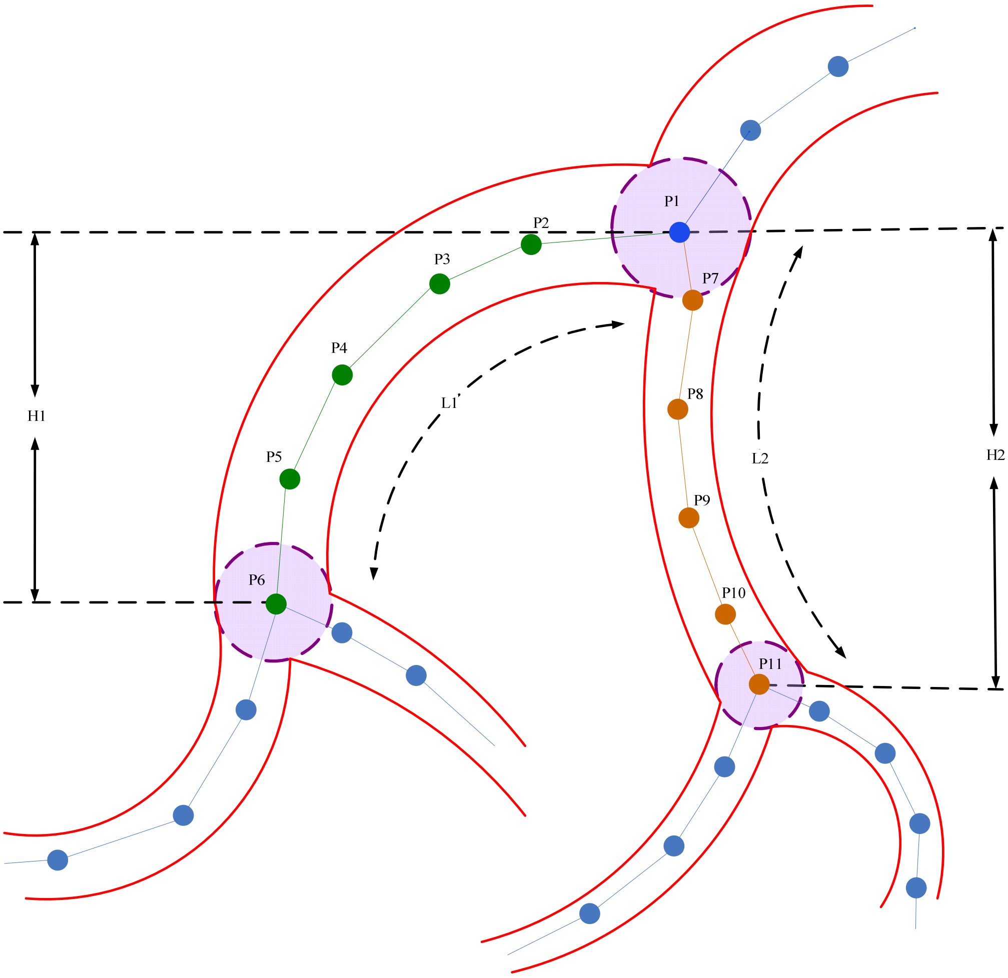


Figure 4.12 Computing length and pressure drop of vasculature

Further, pressure drop is calculated from the following equation:

$$\Delta P = \rho \, g \, h \tag{4.1}$$

Where ΔP is the pressure drop, ρ is the density of blood flow, g is the acceleration of gravity, and h is the height between ends of branches. ρ can be inputted by users for various blood flow simulation, and the defalut value is 1.00 g / cm$^{3.}$ g is 9.8 m/s^2. h is calculated by subtracting the scalar y value between two feature points, and the resulting value should use Abs to insure a positive value. In the diagram, the height between P1 and P6 is H1, and is calculated by Abs (P1.y – P6.y), and the height H2 between P1 and P11 is calculated by using this method as well.

The equation 4.2 is the real equation used for calculating maximum velocity of blood flow:

$$\Phi = \frac{\pi(\mathrm{R}+ a)}{8\eta}\frac{|\Delta P|}{L} \tag{4.2}$$

where R is the cross-section radius reconstructed before, a is the empirical data representing the fourth power of radius. η is the viscosity of blood flow inputted by users, and π is 3.141592654. Φ is the maximum blood flow velocity calculated for all the control points selected before, and recorded as one of the scalar values attached to the corresponding points and then given to the centreline file.

4.5 Configuration File design

There are two types of files generated and saved by using the centreline reconstruction tool: one is the centreline file, which is updated on the basis of the raw centreline file, while the other is the centreline configuration file, which is designed mainly to provide information for contrast medium propagation.

- **Centreline file**

Figure 4.13 provides the file format of the updated centreline file. The difference with the raw centreline file (Figure 4.2) is the scalar velocity attached to every centreline points. Further, the number of branches are enormously reduced regarding to the filter, connection and merging algorithms.

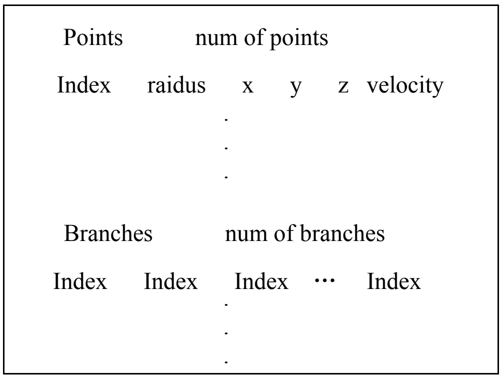


Figure 4.13 File format of the updated centreline file

- **Centreline configuration file**

Figure 4.14 describes the file format of the centreline configuration file. There are three parts in this file. The first part is left for feature points, all the indexes of feature points are recorded, and mapped to the information of the centreline points in the centreline file. The second part is for the feature branches; the number of branches are three times as the number of feature points. The feature branches are recorded on one line, with three row indexes mapping to the branches of centreline file. The third part gives the control points similarly to the feature points. All of the indexes of the control points are recorded and mapped to the indexes of the centreline points in the centreline file.

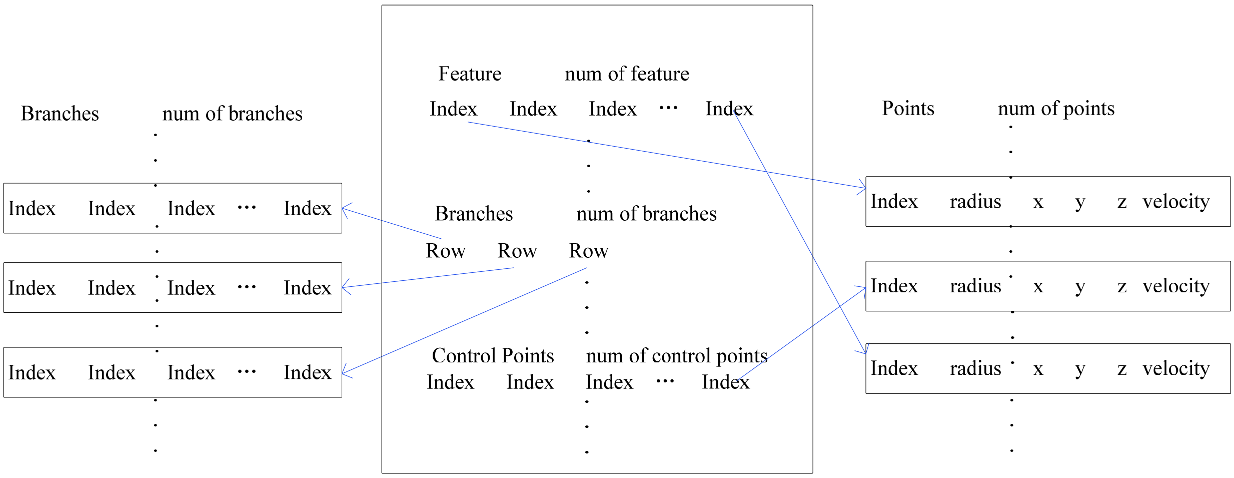

Branches num of branches
Index Index Index ··· Index
Index Index Index ··· Index
Index Index Index ··· Index
Feature num of feature
Index Index Index ··· Index
Branches num of branches
Row Row Row
Control Points num of control points
Index Index Index ··· Index
Points num of points
Index radius x y z velocity
Index radius x y z velocity
Index radius x y z velocity

Figure 4.14 File format of the centreline configuration file

5. Blood flow simulation

This chapter provides details on the chosen model for beating heart. Two key factors on contrast medium propagation are also extracted as forms of linear opacity function and blood flow speed changing function.

5.1 Beating heart modelling

The main factor of blood flow velocity is coming from the fact that the heart beats. The heart of healthy people beats around sixty times per minute, and during the time period of one heart beat, the blood speed is changed from 0 to its maximum value, and returned to 0 again. A simple model for heart beat modelling is to use a sine function as described in figure 5.1, where the horizontal axis represents the time period from 0s to 6s, and the vertical axis represents the value changing from 0 to 1 regarding to the following equation:

$$b \ = |\sin\left(t \times \ \pi \times a\right)| \tag{5.1}$$

Where t is the time point, π is 3.141592653, and a controls the heart rate and varies from one patient to the other. Figure 5.1 describes two kinds of cases using sine function, one with 60 heart beats per minutes, the other with 30 heart beats per minute.

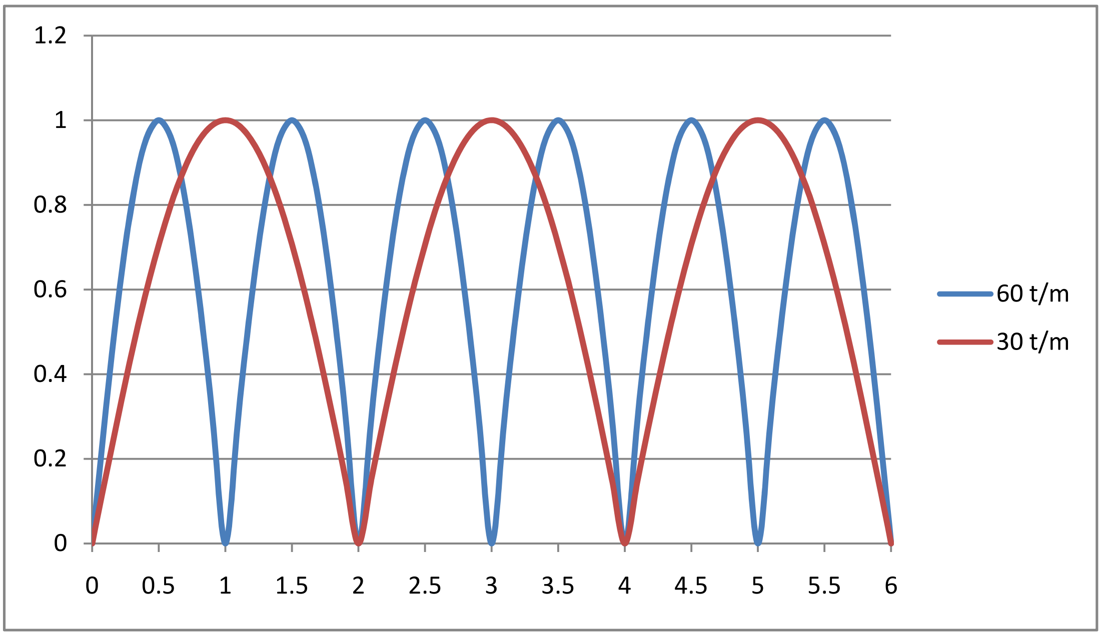


Figure 5.1 Beating heart model with sine function

Different patients have various beating heart rates. It can be approximated by mathematical methods such as the trigonometric function as mentioned before, the Taylor expansion, or other combined functions for modelling realistic beating heart models. The beating heart returns a direct value *b* at each time point during the simulation process, and this value is the key for the velocity calculation, and computing the fading parameter.

5.2 Velocity

The maximum blood flow velocity at each control point is calculated by using the centreline reconstruction tool and Poisseuille Law. The real blood flow velocity at each time point is calculated with the help of both the maximum blood flow speed and the direct value b obtained from the sine function giving the beating blood flow. Figure 5.2 describes the blood flow model at two certain positions with maximum blood speed of 3 m/s and 1.5 m/s. The horizontal axis represents the time period, the diagram shows the time from 0s to 6s. The vertical axis represents the current velocity m/s at each time point between 0s to 6s in the diagram, and which is calculated with the following equation:

$$v = \Phi \times b \tag{5.2}$$

Φ controls the position of the blood flow, which is the maximum blood flow velocity of the nearest control point as calculated before in section 4.5. In the diagram, they are 3 m/s (blue line) and 1.5m/s (red line). b controls the time point of the blood flow, which is calculated from the beating heart model as mentioned before. v is the blood flow velocity at a certain position and at a certain time point calculated by Equation 5.2.

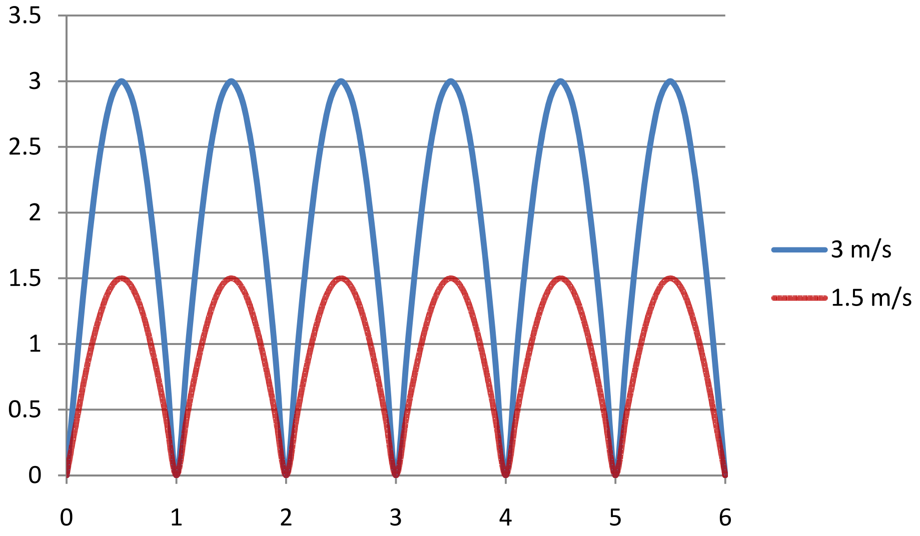


Figure 5.2 Blood flow model at a certain position with maximum blood speed

5.3 Mix Parameter

The mix parameter describes the property of contrast medium dilution in the blood flow. It is given by the following equation:

$$m = v * c \tag{5.3}$$

where m is the mix parameter, v is the blood flow velocity, and c is a constant. The mix parameter is an attribute of the blood flow, thus it is designed in the contrast medium module, and will be added to the final contrast medium attenuation function. However, because of the complexity of the contrast medium propagation and the mixing simulation with the blood flow, so far the mix parameter has not been added to the contrast medium simulation. But the project has now an interface for further development and simulation of a more realistic and scientific interaction between contrast medium and blood flow.

6. Contrast medium simulation

An innovative full set of models of contrast medium is described in this chapter, including propagation modelling, bifurcation and collision detection and response, as well as specific systems of radius-changing, fading, and real-time speed control. The information presented in Chapter 4 and Chapter 5 is combined and used in this set of models and systems.

6.1 Contrast Medium Modelling

The solution I chose for approximating the contrast medium is to use a discrete set of small particles (Figure 6.1). This kind of rendering method is called particle-based rendering. Each particle has a mass, a position, a velocity, a colour, a temperature, a shape and a lifetime [2]. In this project, mass and temperature can be regarded as constant value for all particles, while other parameters are important for contrast medium initialization, speed control and propagation, as well as radius-changing, fading and mixing.

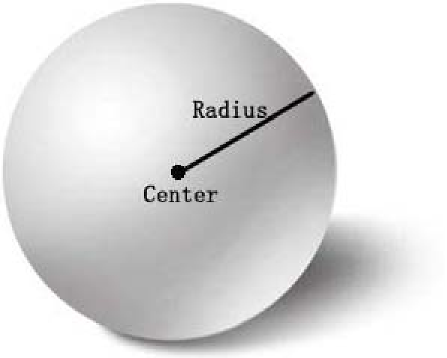


Figure 6.1 Particle

- **The position** of a particle in three dimensions is given by three float numbers (x, y and z) and defines the particle centre. The basic idea of contrast medium propagation is to change the position of particles.
- **The velocity** of a particle includes direction and value of speed. The direction is modelled as a normalized vector d, and the speed is a float number v. According to Newton's second law, acceleration value a is equal to F over m. Thus, if the initial speed of a particle is v_0, its speed value v_t can be obtained after a little time step Δt.
- **The colour** of a particle is black for the contrast medium, and the fading and mixing effect can be achieved by reducing the opacity property of the particle, after a time step Δt. The opacity is obtained according to the opacity formula, which is derived from empirical data given by surgeons, from the blood flow mixing parameter as mentioned before, as well as by the effect of the initial particle speed.
- **The shape** of a particle is related to the parameter radius R, and by changing this parameter, the contrast medium acquires the capacity of fully filling the different sizes of cross-section of the vasculature.
- **The lifetime** of particle describes the time between the creation and deletion of a particle. During the contrast medium simulation, the lifetime influences the process of opacity, changing it from *ini_opacity* to *zero*, according to the opacity formula.

6.2 Contrast Medium Propagation Modelling

After injection from the catheter, the fluid of contrast medium is modelled as a particle line, which is gradually extended dynamically and in real-time with the speed control system. If the contrast medium stays in one vessel during its lifetime (i.e. before fading), keeping one particle line is enough. Otherwise, new particle lines should be created at each bifurcation. This will be discussed in more detail below. The system is designed for infinite recursion so that it can adapt to an infinite number of bifurcations as would happen in a real vasculature.

Figure 6.2 illustrates the particle line initialization and propagation. Simple control points X0, X1, X2, X3 and X4 form the tip area of the catheter, while points after X5 form the body of the catheter, which is not influencing the fluid simulation. The catheter structure and movement are detailed in [1]. P0 and P1 are positions of two particles which are created when the contrast medium is injected from the tip of the catheter. P0 is always created at the position of control point X1, and its radius is very small so that it will not be located outside the vessels. P1 is created at the position of the centre of the cross-section located in the same plane of X0. The radius of particle P1 is initialized according to the radius of the cross-section at that point to fill the vessel. Note that the centre is the same as X0 in Figure 6.2, but it is not always the case. R1 is the cross-section radius initialized by the fluid radius system, described in section 6.5, and which makes the particle (at Position P1) able to fill the vascular cross-section.

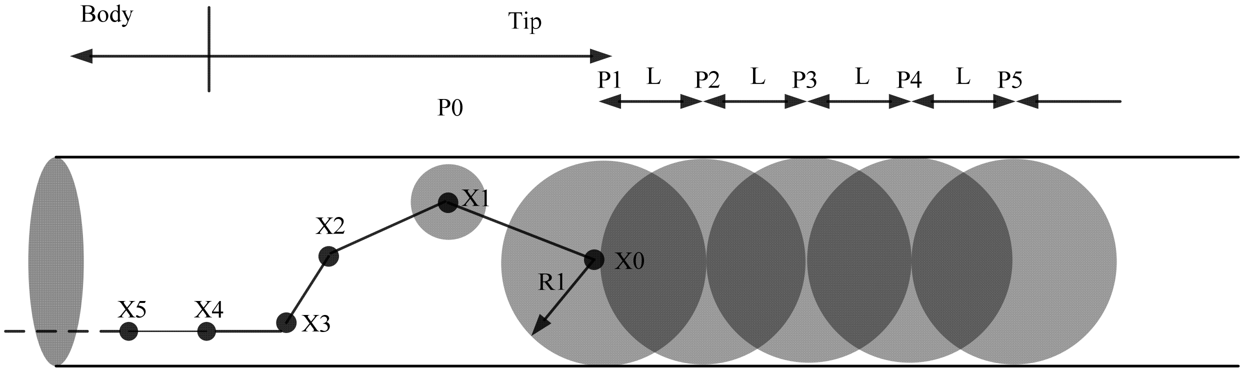


Figure 6.2 Particle line initialization and propagation

P2 is rendered after the time step Δt. It is not a new particle, just the new position of first particle (at Position P1 before the time step Δt). This new position P2 is controlled by three forces, *Fd*, *Fb* and *Fe*.

The force *Fd* represents the injection power, and its normalized direction *d* is obtained from the previous position (P1 in this case) and the first particle P0. The magnitude of *Fd* is related to the average injection speed *v*. In the simulation, the injection speed *v* is used rather than using *Fd*, since the VSP haptic device can detect the injection speed every time frame.

The force *Fc* originates from the response of the collision detection, discussed in section 6.4. Its direction is given by the average normal *n* of the colliding vascular triangles, and its magnitude is related to the distance *d* from the closest colliding triangle, the smaller the distance, the bigger the magnitude.

The force *Fb* represents the motion of the blood flow. Its direction *d* is the same or opposite to the direction of *Fd*, depending on whether the contrast medium propagates anterogradely (i.e. towards the leg) or retrogradely (i.e. towards the head). The magnitude of *Fb* is related to the magnitude of velocity obtained from the blood flow module as described in chapter 5.

Figure 6.3 describes the force analysis of three forces. Since *Fc* can be considered as perpendicular to the internal edge of the vasculature, it theoretically has no effect on the contrast medium. Thus, the motion of fluid propagation is controlled by *Fd* and *Fb*, and is maintained by the velocity control system which will be described in section 6.6.

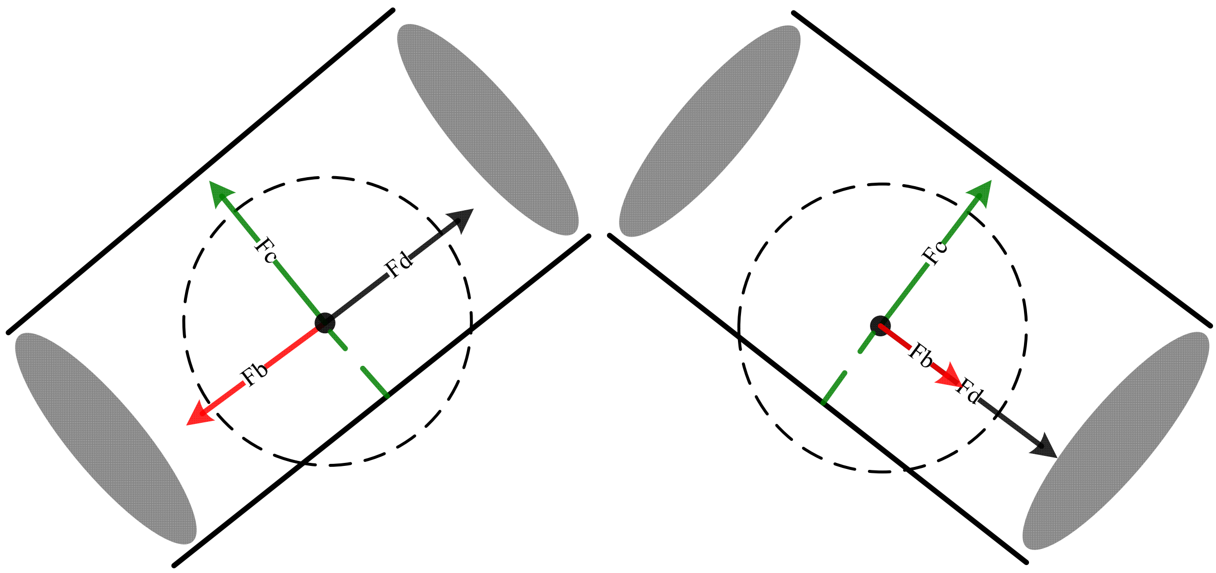


Figure 6.3 Force analysis of a Particle just injected. The contrast medium propagates retrogradely (left), anterogradely (right).

P2 is kept in the vessel and is localized near the centre of the cross-section. All the previous positions of each particle are stored in a linked list, to get a continuous fluid line effect. In figure 6.2, the particle would move forward length L during each time step from P1, P2, P3, P4, P5 etc. Moreover, the radius of this particle is controlled by an adaptive radius system (see below section 6.4), which dynamically modifies the size of the particle radius to fill the vessel as best as possible.

6.3 Collision Detection and Response

Based on the real-time collision detection system in the existing framework, the particle line is constrained within the vasculature. This system is explained in detail in the previous work [1] and the main idea is to use an Axis Aligned Bounding Box (AABB) tree to partition the vasculature and a bounding sphere around each contrast agent particle. The AABB tree is created only once: when the vascular model is loaded, before the simulation starts. This partition method aims at having a relatively equal number of triangles in all leaves to speed up the search time by having a balanced tree [1].

The collision response computes the external force applied by the vasculature on the colliding particle(s). This force is calculated by the average normal n of the colliding vascular triangles and the distance d from the closest colliding triangle. With the help of the collision detection and response systems, all the colliding particles are constrained inside the vasculature.

6.4 Fluid Radius System

This system is divided into three parts. It includes computing the initial cross-section radius, adapting the particle radius during propagation, and solving abnormal cases like large aneurysm with shape radius change.

- **Computing initial cross-section radius**

This part is responsible for computing the initial cross-section radius of the particle (P1, figure 6.2). Figure 6.4 illustrates the process of computing with two different cases. The green spheres with bounding dashed lines are control points obtained from centreline configuration file. In the diagram, RP2 is near one of the feature points (also read from centreline configuration file). In this case, the cross-section of RP2 will be interpolated from the three neighbour control points of the feature point with equal weights. The feature detection method will be discussed later. RP1 is not near one of the feature points, thus the cross-section of RP1 will be interpolated from the nearest two control points with equal weights.

Since the vasculature has multiple branches, the radii of the cross-section change irregularly, and thus the particle radii should be adapted in order to completely fill the vessel lumen. When the fluid meets a bifurcation and the cross-section changes apparently, control points are still used for re-computing the initial cross-section radius.

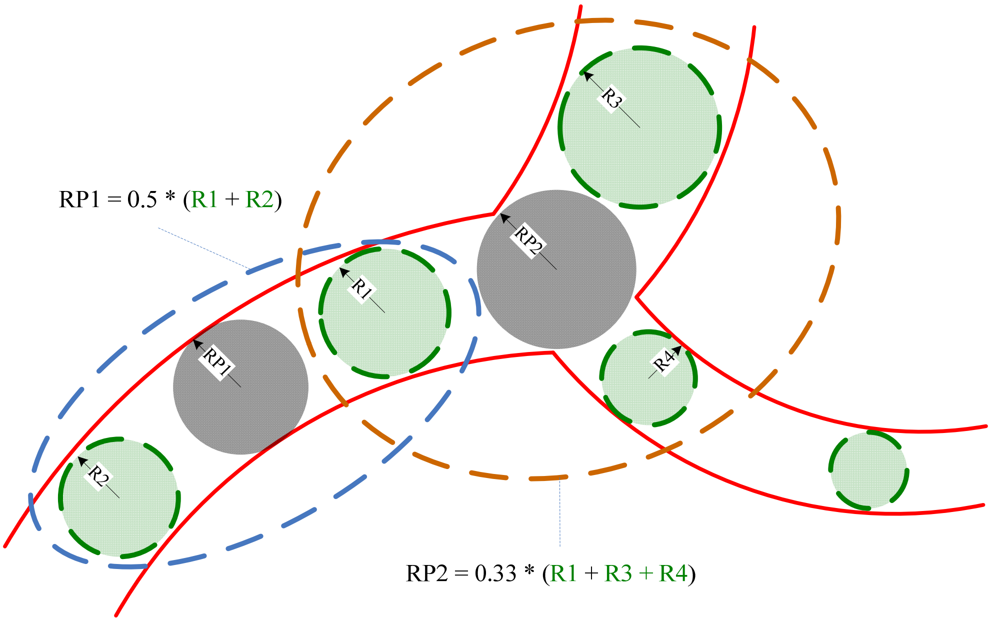


Figure 6.4 Computing the cross-section of the initial particle using control points

- **Adaptive fluid radius changing**

In one branch, the radii of the vascular cross sections do not normally change sharply, even though some lesions of vessels (such as occlusion and stenosis) could change the radii of the cross-section in some places. Since this feature should be seen under diagnosis, the contrast medium should also fill these three dimensional areas. Because the cross section radii are not known yet, a simple solution has been implemented. The particle radius is increased gradually at each time step *t* if that particle failed to collide with the vessel walls. On the other hand, the particle radius is reduced gradually at each time step if a collision is detected. Figure 6.5 describes this process, and radius position P2, P3 and P8 and P9 are two cases for radius increase and reduction.

When the particle is at position P2 with radius R2, it does not collide with the vasculature, thus the radius is increased by dr1, and the radius of the new position of the particle becomes R3. Although it does not fill the cross-section perfectly, a small change can still be performed. When the particle is at position P8 with radius R8, it collides with the vasculature, thus the radius is decreased by dr2, and the radius of the new position P9 of the particle becomes R9. Thus, by always using this system, the fluid can always nearly fill the cross-section of the fluid in the normal situations.

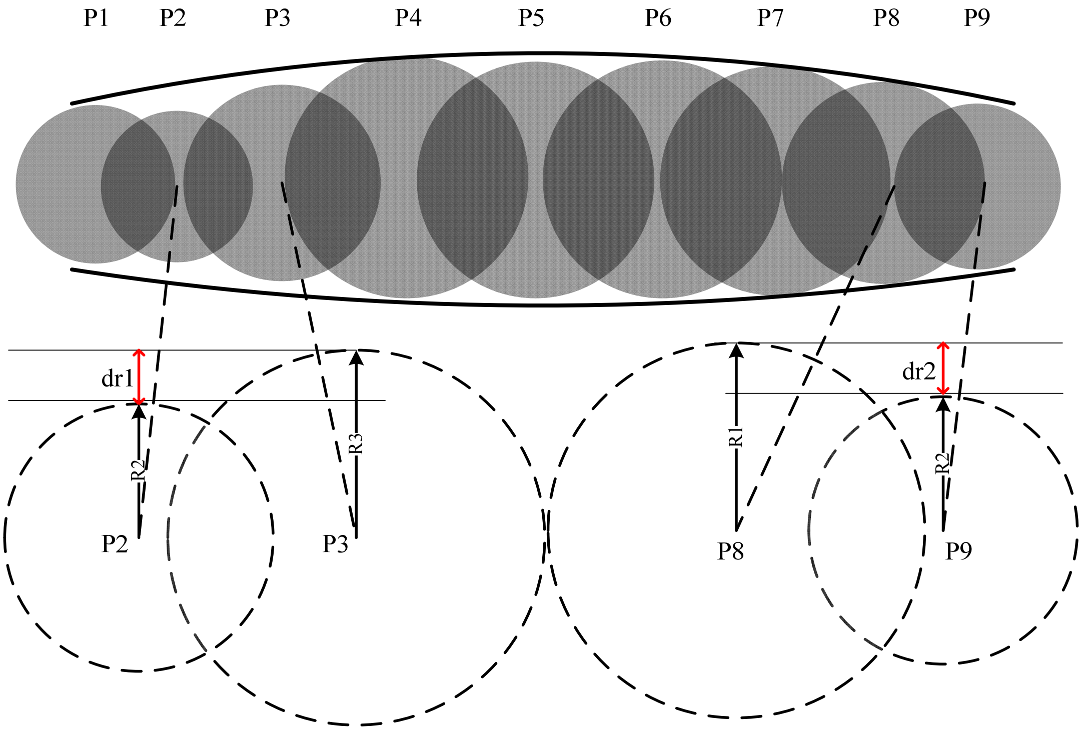


Figure 6.5 Adaptive fluid radius changing system

- **Abnormal cases**

In one branch, abnormal cases like huge aneurysm with sharp radius change are not well shown if using the precedent adaptive radius changing system. Thus, this kind of feature point should be marked before by the centreline reconstruction tool, and interpolating the control points when meeting this kind of feature. This feature should be developed later as it was not implemented in the current project.

6.5 Fluid mixing and Lifetime Control

A drawback of the particle line design described in section 6.2 is that only the first particle can be controlled at each time point. Another region of fluid thus needs to be created when meeting the blood flow and mixing with it, this is done by using a second particle line. Therefore, a two particle line model is designed and implemented for this aim. Through life control, the fading system affects all the particles of a particle line and creates the effect of mixing between two fluids with different properties, consequently enhancing the realism of the simulation.

- **Two particle line model**

From the retrograde initial injection, the fast moving blood flow should mix with the contrast medium directly and then propagate anterogradely. To simulate this phenomenon, a two particle line model is introduced. Figure 6.6 describes the two particle line model. The principle is the same as illustrated in Figure 6.2: X0 and X1 are still the two positions which are related to the first two particle initialization.

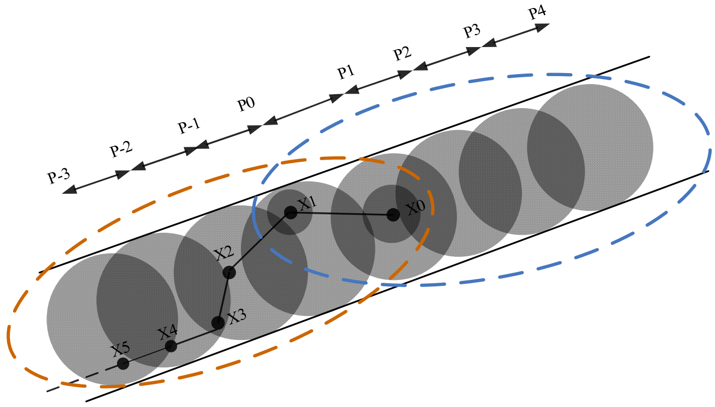


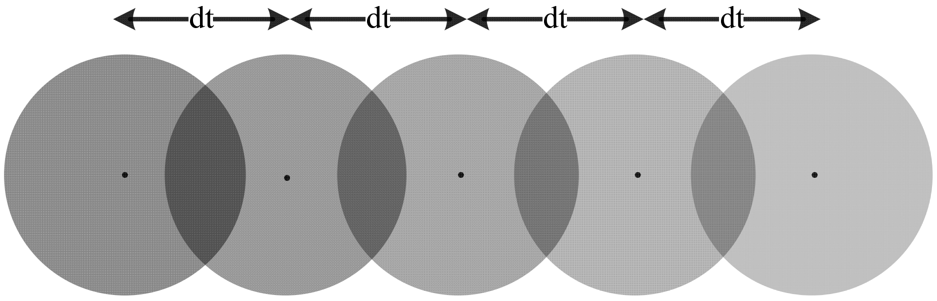
Figure 6.6 Two particle line model

In figure 6.6, the first particle line (P0, P1, P2, X3) uses catheter particle X1 as the position to create the small particle, and uses catheter particle X0 as the position to create the large particle. The second particle line (P0, P-1, P-2, P-1) uses catheter particle X0 as the position to create the little particle, and uses catheter particle X1 as the position to create the large particle, however, as mentioned in section 6.1, the position of the particle is not always in X1, the centre of the particle is near the centreline of the vasculature because of the effect of the force *Fc* from collision response.

- **Fading system**

From the initial injection, the concentration of contrast medium in the blood flow is high and thus, the appearance of the contrast medium under X-ray is deep black at that moment. However, the contrast medium will gradually mix with the blood and finally it would pass with the anterograde blood flow down towards the legs. Further, the velocity of both the initial injection and of the blood flow speed at each passed point also affects the mixing effect. Besides, a large quantity of contrast medium injected stays longer than a small quantity of contrast medium.

Under X-ray, the medical image would show that the contrast agent colour gradually becomes weaker and weaker, and finally totally disappears. In consequence, the fading system is developed with a linear opacity function (Figure 6.6), which combines the consideration of mixing parameter obtained from blood flow module *m*, quantity of contrast medium injected *q*, and average injection speed *v*. *do* represents the changing value of opacity during each time step *dt*. The constant values *a*, *b* and *c* can be specified for various kinds of contrast medium. Among others, the injection speed *v* is positive if the fluid runs anterogradely, or negative while it runs retrogradely.



Linear Opacity Function: $do = am + bq + cv$

Figure 6.6 Linear opacity function

With these factors, each particle colour depth gradually reduces with time along the particle line if the fading begins.

- **Lifetime control**

The lifetime control system is responsible for managing the start time and concentration of fading for each particle during the contrast medium propagation. Through the lifetime control system, each particle of the simulated contrast medium is affected by a realistic mixing effect related to various opacity values.

The particles within retrograde contrast medium begin to fade away when the injected contrast medium arrives at the maximum quantity of injection. The opacity of each new shifted particle is attenuated by a value u. For example, if the opacity of the particle at P1 is 0.5, then at P2 is $0.5 - u$, at P3 is $0.5 - 2u$.

The particles within anterograde contrast medium begin to fade away when the opacity of all the retrograde contrast medium is equal to zero, and attenuation value u is not used in these particles since the blood mixed contrast medium propagates anterogradely constantly until the above contrast medium is totally faded away (i.e. fully mixed with blood flow).

6.6 Velocity control system

In a standard angiography, a needle is first inserted in the vasculature, usually in the femoral artery. Then, a guidewire and a catheter are navigated inside the vessels. To visualize the vessels, contrast agent is injected. It propagates rapidly retrogradely (i.e. towards the head) along the vessels, including each common iliac artery at around 6 ml per second.

As mentioned in figure 6.3, *Fd* can be considered as a momentary force which gives the initial velocity to the injected particles, while *Fb* is a sustained force which changes the velocity of contrast medium during the whole process of propagation. Thus, acceleration *a* is described as the following equation:

$$a = \frac{Fb}{m} = \frac{dv}{dt} \tag{6.1}$$

where m is the mass of the particle, dv is the rate of magnitude change of a particle at each little time step dt. In the simulation, *Fb* is just used for motion analysis since the force is not easy to obtain. However, the magnitude of the blood flow at different time points and at various positions of the vasculature can be obtained from the blood flow module. For example, at a position P, the blood flow vb_{t1} is obtained at time point *t1*, and vb_{t2} at time point *t2*. Thus the acceleration a_{t2} yields to:

$$a_{t2} = \frac{vbt1 - vbt2}{t1 - t2} \tag{6.2}$$

Since the time period between *t1* and *t2* is equal to the rendering time frame, it can be considered as a differential element of time. a_t is changed dynamically regarding to the beating heart model as described in chapter 4, and it changes the magnitude of the contrast medium velocity by using the following equation:

$$v_{t2} = v_{t1} + a_{t2} \times (t2 - t1) \tag{6.3}$$

Where v_{t2} is the velocity magnitude of a particle at position P at time point *t2*, while v_{t1} is at time point *t1*, a_{t2} is nothing but the acceleration of the blood flow at the same position P, with a positive or negative direction. The boundary values of v_{t2} are from zero to the maximum value of the blood flow. Whenever it reaches the boundary, it propagates with the same velocity as the blood flow at each time point. Figure 6.7 describes the moving track of a particle under the effect of *a* without the effect of the collision detection (the translation is always in a pipeline which is horizontal with no radius change). Diagram (a) shows the particles propagating retrogradely, the positions of the particles are changed regarding to the direct velocity at each time point from *t1* to *t5*, and the direct velocity is also changed regarding the velocity acceleration. For example, in figure 6.7, the particle in *P3* at time point *t3* is ready to propagate of *Shift* distance. *vt3* is known from the previous calculation, and *at3* is calculated from the beating heart model and the equations are described in the diagram. By using Euler's Law, *Shift* can be calculate by $vt3 \times (t4 - t3)$, and by using Newton's Law, *vt4* can also be calculated by $vt3 + at3 \times (t4 - t3)$. Among others, *vt4* is constrained between 0 and initial injection speed *v0*. When the velocity reduces to 0, then the contrast medium should change direction, while in the real simulation, time control system would release the signal of fading rather than really changing the fluid direction.

Diagram (b) shows the particles propagating anterogradely, the basic idea of changing velocity is the same as described in diagram (a). The only difference is that the velocity in this case is constrained within *v0* to the maximum blood flow value, where the value *v0* can be zero (if the particle line is the second particle line for blood mixed contrast medium simulation), or it can be the initial injection velocity.

The maximum blood flow value is obtained from the centreline file attached to the nearest centreline point where the injection happens. If the particle velocity reaches the maximum blood flow velocity, then the velocity speed follows the same changeable rule as the blood flow (i.e. read directly from the beating heart model).

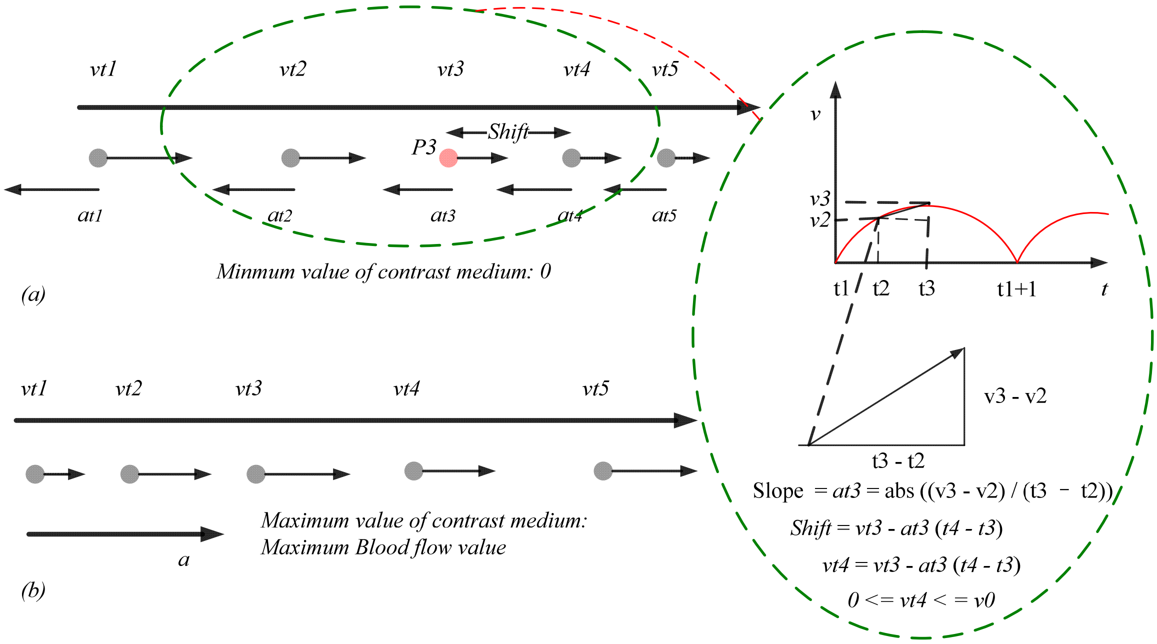

vt1
vt2
vt3
vt4
vt5
Shift
P3
at1
at2
at3
at4
at5
Minmum value of contrast medium: 0
(a)
v
v3
v2
t1
t2
t3
t1+1
t
v3 - v2
t3 - t2
Slope = at3 = abs ((v3 - v2) / (t3 – t2))
Shift = vt3 - at3 (t4 - t3)
vt4 = vt3 - at3 (t4 - t3)
0 <= vt4 < = v0
vt1
vt2
vt3
vt4
vt5
a
Maximum value of contrast medium:
Maximum Blood flow value
(b)

Figure 6.7 The moving track of the particle with the effect of velocity acceleration is in the opposite direction as the blood flow direction (a), in the same direction as the blood direction (b).

Figure 6.8 illustrates the acceleration curve of a patient whose beating heart is modelled using the sine function with a rate of 60 beats per minute and a maximum blood flow speed of 3 m/s. By using this curve, in one branch of the vasculature, the acceleration of the contrast medium at each time point can be directly read from the curve. Thus, it is simple to calculate the velocity magnitude at next time frame.

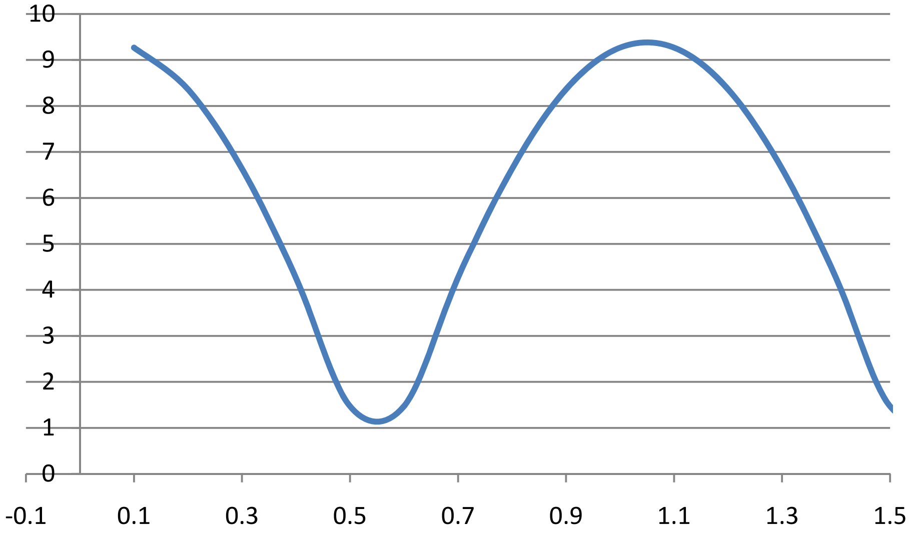


Figure 6.8 Acceleration curve

Figure 6.9 illustrates the fluid propagation with the effect on the acceleration from *Fb*, and the initial speeds from *Fd* and *Fc* obtained from the collision response.

The direction *dir* of fluid velocity is simply calculated from the vector between the two initialized particles, one is always fixed with the second point of the catheter, and the position of the other is changed all the time. *Vf* is the vector (*fx, fy, fz*) which is added to the position (x, y, z) by the effect of *Fb* and *Fd,* and it is computed via the following equation:

$$Vf = v \times dir \qquad \textbf{(6.4)}$$

where *v* is the velocity magnitude as described before, and *Vc* is the vector (*cx, cy, cz*) which is added to the position (x, y, z) by the effect of *Fc*. The computing equation yields to:

$$Vc = dc \times n \qquad \textbf{(6.5)}$$

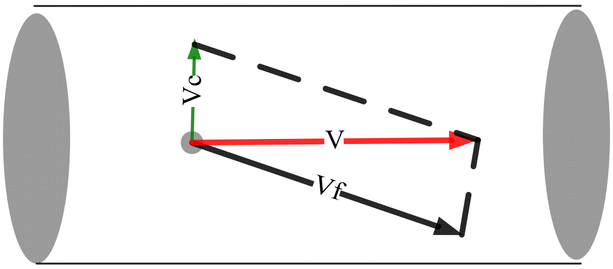


Figure 6.9 position vector resolutions

where the average normal *n* is the one from the colliding vascular triangles and the distance *dc* is from the closest colliding triangle as mentioned in section 6.3. *V* is the final vector (*vx, vy, vz*) combined by *Vc* and *Vf*. This vector controls the shifted position at one time frame period, which considers both the fluid hydrodynamic theories and the vessel boundary forces. With the help of this vector, the contrast medium propagates realistically, and always follows the path of the vasculature. The equation of the changed position ΔPos yields:

$$\Delta Pos = V \qquad \textbf{(6.6)}$$

6.7 Bifurcation Detection and Response

Based on the models and theories described in the previous sections, the contrast medium propagates realistically and in real-time through a branch of a vasculature. However, in a complex vascular system, bifurcation modelling is necessary to establish how the contrast medium propagates through all branches when meeting a bifurcation point. The model includes two parts: bifurcation detection and bifurcation response.

6.7.1 Bifurcation detection

Figure 6.10 describes the bifurcation detection model. The feature points are obtained from the centreline configuration file and generated and localized by the centreline reconstruction tool. When the first particle is propagating with dynamic changeable velocity and opacity, it checks for the bifurcation detection system. The basic idea is controlled by an equation as illustrated in the diagram. Where d is the distance between the first particle and all the bifurcation points in the whole vasculature system. *shift* is the magnitude of the changing position vector | ΔP|. If d is equal or less than shift, then it means that a bifurcation is detected, and the *switch* variable on that bifurcation is set to 1. The use of *switch* is to protect the feature from being detected again and again, as a particle line is designed to meet only one bifurcation point as new lines are created at bifurcations, while the current particles disappear as explained in next section.

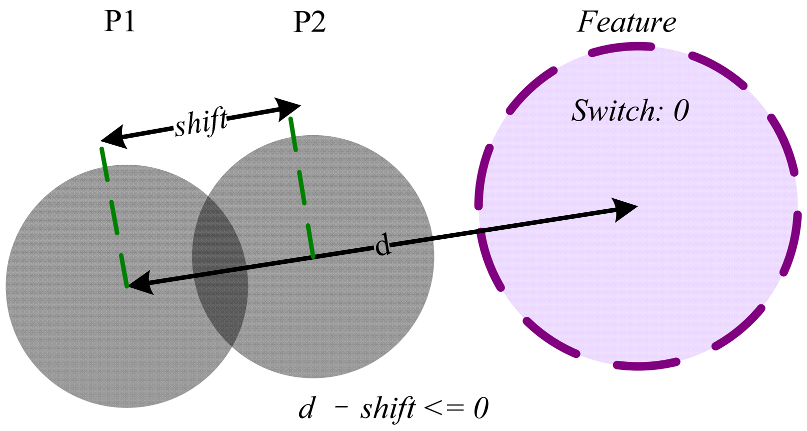


Figure 6.10 Bifurcation Detection

6.7.2 Bifurcation response

Compared to bifurcation detection, the bifurcation response is more complex. In figure 6.11, P2 is the bifurcation point and it is detected by the bifurcation detection system. Around P2 there are three branches, and P3, P1 and P6 are neighbouring branches of P2 (the nearest centreline point in each branch). The contrast medium direction F is illustrated in the diagram. The contrast medium is coming from branch A, and two new particle lines will be created and propagated through branch B and branch C, respectively. The response process is divided into three parts: firstly, the *original branch detection* is to identify which branch is the original branch through which the contrast medium is propagating. Secondly, *data return* transferring useful bifurcation information for creating new particle lines. Lastly, an *infinite particle line creation system* starts with various parameter initializations.

- **Original branch detection**

The original branch detection involves a series of vector analysis, and it is illustrated in figure 6.11. In the vector analysis circle, d1 is the normalised vector from P1 to P2, d2 is the normalised vector from

P3 to P2, and d3 is the normalised vector from P6 to P2, where P2 is the detected bifurcation vector and P1, P3 and P6 are neighbour vectors which are all read from centreline configuration file. F is the fluid direction vector from the first particle of contrast medium to P2. By calculating the vector magnitude between d1, d2, d3 and F, respectively, the minimum value of the three is recorded and the related branch can therefore be detected regarding the corresponding bifurcation branch information read from the centreline configuration file. The algorithm is as follows:

d1 = normalize (P1 – P2)
d2 = normalize (P3 – P2)
d3 = normalize (P6 – P2)
c1 = |d1 – F|
c2 = |d2 – F|
c3 = |d3 – F|
c1 = min (c1, c2, c3)
c1 is related to d1, d1 is related to P1, P1 is related to Branch_A (from branches information of configuration file)

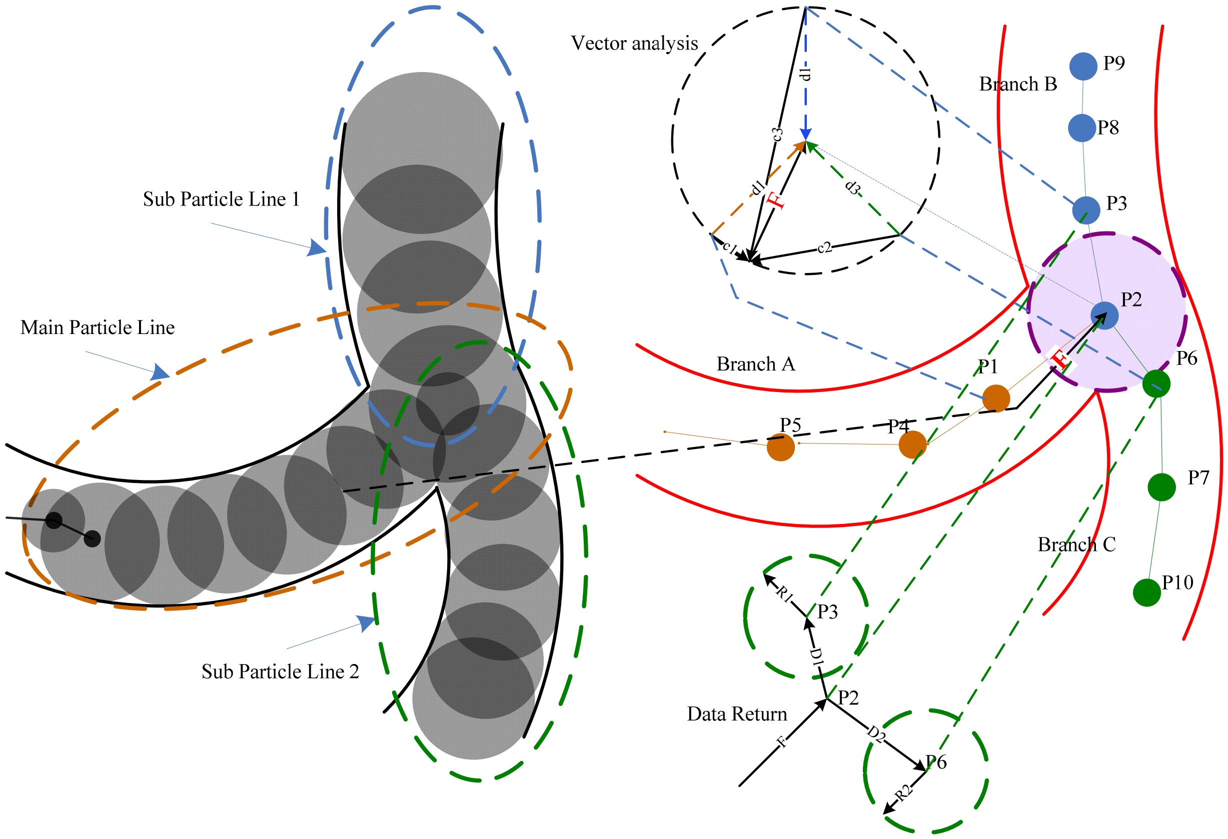


Figure 6.11 Bifurcation Response

- **Data return**

The bifurcation response returns basic information for new particle line creation, including the centreline points P2, P3 and P6, and the cross-sections R1 and R2 of P3 and P6, respectively. From these three sets of data, two particle lines can be initialized. The sub particle line 1 is initialized by creating a small particle at position P2, and creating another particle with radius R1. The sub particle line 2 is initialized by creating a small particle at position P3, and creating another particle with radius R6. The basic idea is similar to the initialization of the contrast medium injection. D1 is the initial direction of the sub particle line 1 propagation, while D2 is that for sub particle line 2.

- **Infinite particle line creation system**

With the mechanics of original branch detection and data return, particle lines can be created infinitely for dealing with complex bifurcations of real vasculatures. In the implementation and programming, the initial particle line is maintained in a class, and in the main class, there are two sub objects of the self class. With the help of recursion, the infinite particle line creation system is established. Furthermore, it needs to create a realistic smooth connection between different particle lines in the fields of opacity, velocity and quantity, etc. Thus, these parameters are also transferred to the sub initialized objects of the particle line class. Moreover, the initial maximum blood flow value is obtained from the neighbour points of each branch, and then the changing speed is maintained by the velocity control system as usual.

7. User-interaction

Users can inject contrast agent multiple times in the catheterization training process. Furthermore, the contrast medium can be injected through the VSP interface or with the keyboard and the user interface. A control panel is designed for parameter control over the contrast medium. Different patient cases have been generated to widen the scope of the training, and a simulation review system (based on 3D snapshots) can be used for evaluation and tracking for the whole process of the contrast medium injection.

7.1 Infinite-injection system

In a complete angiography, the contrast medium would be injected several times and at different places. Thus, the possibility of an infinite number of injections has been developed. In this system, each time of injection is isolated from each other, it initializes one or two particle lines with necessary parameters like average injection speed, maximum blood flow velocity value, and catheter position, as well as quantity of injected contrast medium, bifurcation and feature points of the vasculature, and so forth. Moreover, each time of injection has its own life control system and fading system, isolated velocity control system, and deleting mechanism when the particle line is totally faded away. Thanks to an efficient encapsulation, the infinite injection is possible. With the help of the deleting mechanism, the rendering speed is satisfying since the faded particle lines will be released and will leave memory available for new visible contrast medium injection. The infinite injection system greatly improves the user interaction, and the injection behaviour can be achieved using the VSP interface, the keyboard, and the control panel as described below.

7.2 VSP device

The VSP device has an interface that can be connected with a VSP syringe. Through the syringe, air can be injected rather than real fluid. Figure 7.1 describes the VSP syringe states at two time points t1 and t2. At time point t1, users operate on the plunger by applying a force *Fd*, and let the plunger produce an initial speed *v*. The initial speed *v* can be detected by the VSP syringe sensor and, through the VSP interface, the sensor returns the speed value to the simulator. It can then be used by the framework. From the time being, the air between the pipe connection and the front of the plunger is injected gradually in the VSP. During this process, the only value that can be detected is the speed *v* as mentioned before. At time point *t2*, all the air is completely injected. Although the speed *v* can be obtained at each time frame, the time frame is not perfectly stable thus the model is simplified by using the following equation:

$$q = v \times (t2 - t1) \times \pi r^2 \qquad \textbf{(7.1)}$$

where *q* is quantity of air (i.e. the virtual contrast medium), *v* is the initial speed (the injection process is considered as the motion of constant velocity), *t2* and *t1* are the two time points, and πr^2 describes the area of cross-section of the syringe.

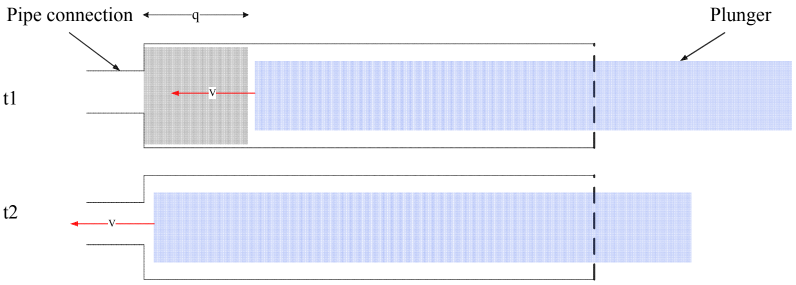


Figure 7.1 VSP syringe states at two time point t1 and t2

7.3 Different ways of injection

- **Control Panel**

The diversification of multi-injection is modelled as a user-interface input panel developed in FLTK that will be discussed later in chapter 8. Three parameters can be tuned: particle numbers, speed of contrast medium propagation, and blood flow speed. These values are obtained by a call back function and stored in arrays integrated in the structure created for each lifetime injection. They can therefore be obtained and added into different models and equations as mentioned before. The data is totally segregated from the detailed implementation models and equations. Further parameters can be easily added thanks to the expansibility of call back functions. More complex medical simulations can be achieved with an improvement of the models and equations.

The blood flow speed can be changed by users. However, in most cases, the blood flow velocity speed is read from the centreline file through complex hemodynamic calculation.

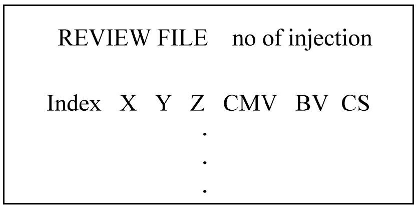
- **Keyboard**

Besides using the control panel, the keyboard provides a simple way for injection; the space bar is designed and implemented for contrast medium injection. The keyboard is controlled by the input manager, and keeps listening to the change of states of the keyboard. The injection command is sent out whenever the space bar is pressed, and default parameters such as contrast medium quantity, initial fluid speed, and initial opacity are used. All the systems and models are cooperating for completing one injection.

7.4 Simulation review system

This review system provides the evaluation function for the contrast medium simulation, and it will further be added into the simulator review module combined with other evaluation programs. Users can use the user interface to choose four stages throughout the whole injection process just like an interventional radiologist would use the video recording system after a contrast injection. The implementation is achieved by storing the vtkActors of all the particle lines in the graphic engine. The behaviour of the contrast medium (throught its vtkActors) is automatically stored at four times during the whole lifetime of one injection. The time location is calculated by the pre-estimation from contrast medium quantity and initial injection.

The review of the contrast medium injection then displays the stored vtkActors through the graphic engine. This review system, also called road map or Digital Subtraction Angiogram (DSA) is very useful for the interventional radiologists as it allows them to see where the contrast agent propagates and therefore where the vessels are without injecting more contrast medium in the patient (which could damage his/her kidneys). This is a key step in the intervention as it permits to reduce the dose of contrast agent injected while still giving a clear map of the vessels while the clinician navigates in them with the medical instruments.



```
REVIEW FILE    no of injection

Index  X  Y  Z  CMV  BV  CS
                .
                .
                .
```

Figure 7.2 Review file format

Besides the graphic review, the positions of each particle of the contrast medium is recorded and written into a review configuration file. This file also includes the contrast medium velocity, blood flow velocity and vasculature cross-section information of those recorded positions. The file format is described in figure 7.2, and reviewers could use this file as a useful basis for evaluation.

8. Evaluation

This chapter presents an evaluation of the contrast medium simulation as well as the performance of the centreline reconstruction tool, and gives an example with different patient-specific vascular cases. Further, feedback from clinicians and comparison to real injection videos is given for evaluation of the simulation.

8.1 Performance

The result of both centreline reconstruction tool and contrast medium simulation are presented here in relation to the design and implementation described before.

8.1.1 Centreline Reconstruction Tool

The vessel representation as a centreline is a key part of our IR simulator as it is light and efficient and therefore allows real time simulation, while providing the most important features and properties of the vasculatures. Using an accurate centreline as a vessel simplification allows the simulation to be performed realistically and in fast frame speed. Projects such as contrast medium simulation and vessel surface deformation, and other further development in our group, are highly related to the centreline information of vessels or other tubular structures (intestines...). The presented centreline reconstruction tool provides both automatic algorithms and manual ways to achieve a qualified centreline for various vessels. Further, regarding to different projects, various information could also be grasped and stored as forms of configuration files.

8.1.1.1 Control panel and graphic view

Figure 8.1 provides the graphic view and control panel of the centreline reconstruction tool, which is a simple user-interaction system, including two parts:

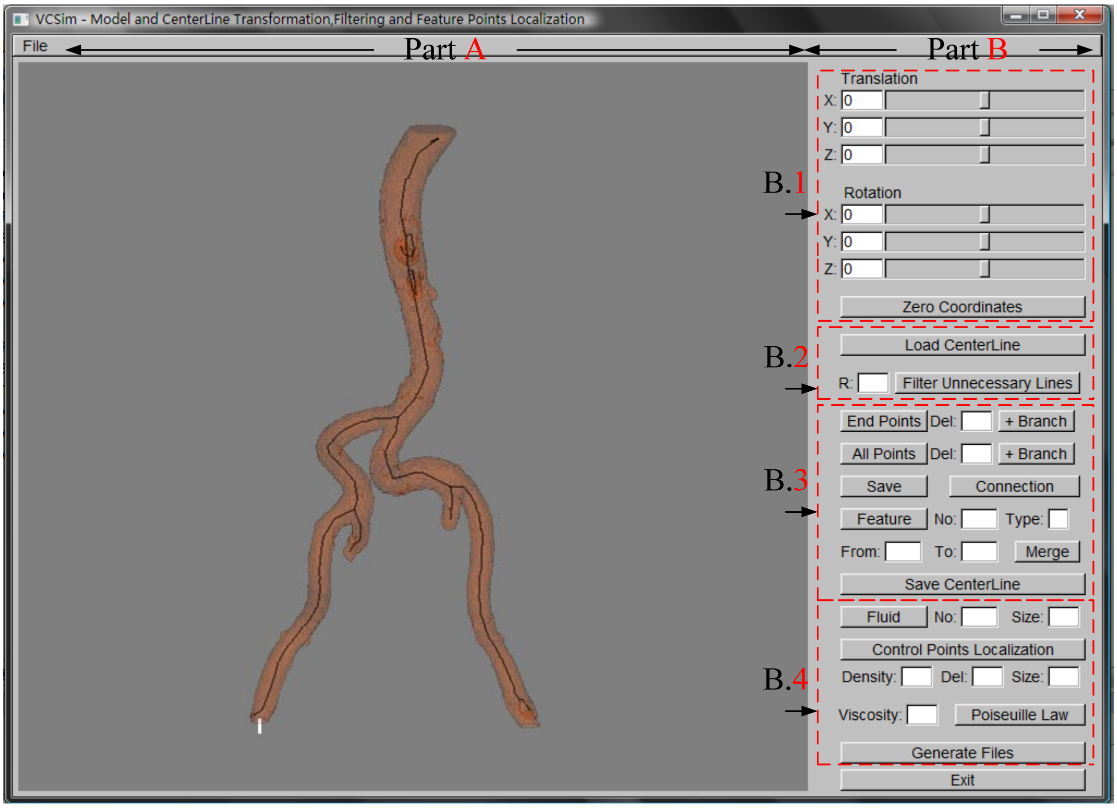


Figure 8.1 Graphic view (Part A) and Control panel (Part B) of the centreline reconstruction tool

Part A is the graphic window. It shows the vessels, centrelines and key information on the vasculature as forms of coloured points. Further, any useful user update can directly be performed on the graphic view. Besides, users can change the view point by using the left button of the mouse (LBM), and change the view distance by using the right button of the mouse (RBM).

Part B is the control panel, including four areas. B.1 is the translation and rotation area where users can specify the transformation of the vasculature to a satisfying position (normally one of the branches is set over the initial position of the catheter, shown in white in Figure 8.1). B.2 is the centreline filtering area where users can first load the centreline from a file, and the centreline would automatically be matched with the updated position of the vasculature. B.3 is the area of centreline reconstruction, combining selection, deletion, connection and merging, and then saving a satisfied centreline for simulator use. B.4 is the area specifically designed for contrast medium simulation. It produces useful information for fluid simulation, such as viscosity and bifurcation information, etc.

The detailed information of how to use this tool is described in the User-Manual. Results of key techniques are described in the next three sections. These are centreline filtering, key point localization and centreline reconstruction. Besides the first one, the other two are repeatedly used when reconstructing a centreline, as well as in designing a standard fluid configuration file for contrast medium simulation.

8.1.1.2 Centreline Filtering

Figure 8.2(a) provides the original centreline generated from the 3D vasculature tool [40], and Figure 8.2(b) shows the centreline after using the filtering function of the tool. About **489** unnecessary centreline points and **174** unnecessary lines are filtered. However, filtering creates a disconnected centreline, and some specific areas are over filtered, thus, the filtered centreline would need to be reconstructed. Besides, one branch (i.e. between the two main bifurcations) should be represented using one centreline, therefore merging is necessary. In order to control the reconstruction process, key points are localized to help deleting necessary points and lines manually, as well as connecting and merging lines.

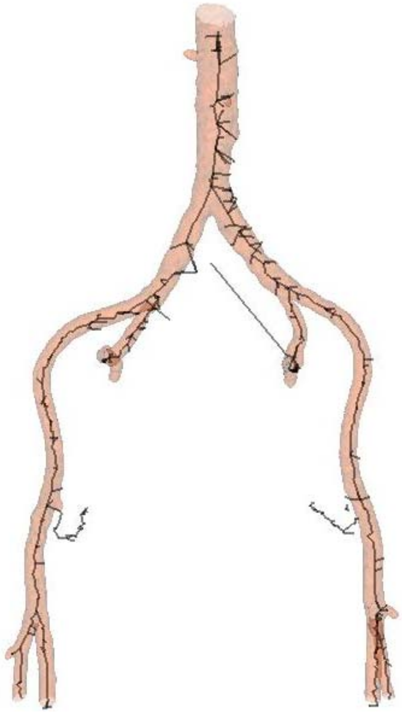


Figure 8.2 (a) Original centreline before filtering: many lines are useless

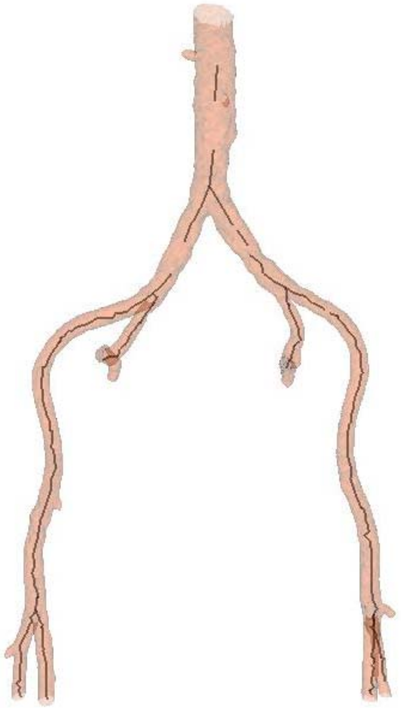


Figure 8.2 (b) Centreline after filtering: disconnections appear

8.1.1.3 Key Points localization

Figure 8.3 describes four kinds of key points, including end points, control points, bifurcation points (i.e. feature points), and feature neighbour points as introduced in chapter 4. They can be used as navigation points (regarding the indexes of points) for centreline point deletion, connection and merging. Besides, by specifying the radii of points, one can also reconstruct the centreline cross-section. Further, the velocity attached to control points can be calculated as described in section 4.4. Thus, after accomplishing these processing steps, the feature control and end points can be written into the centreline configuration file, while the changed radius information and velocity information can be recorded in the updated centreline file.

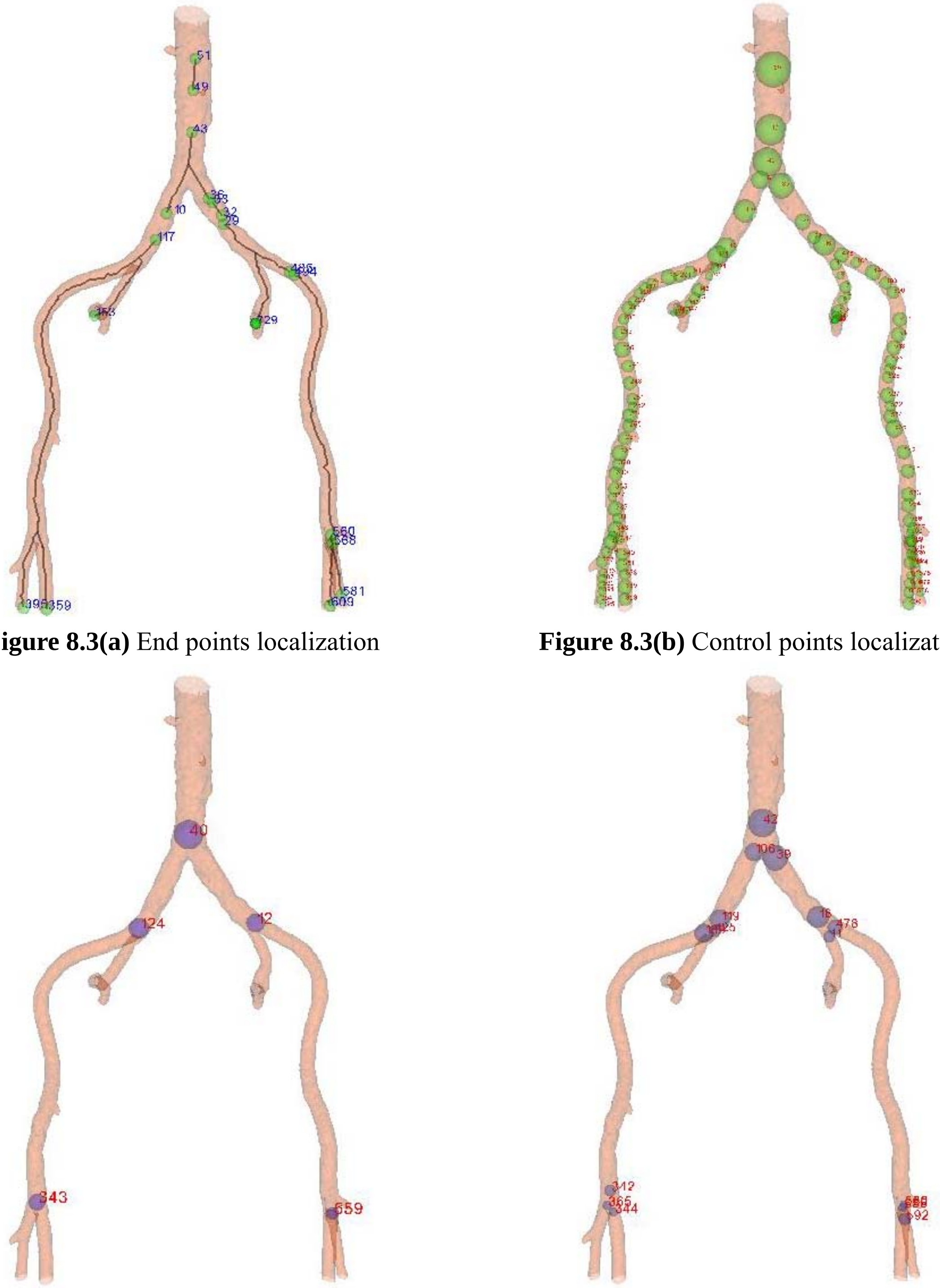


Figure 8.3(a) End points localization

Figure 8.3(b) Control points localization

Figure 8.3 (c) Bifurcation points localization

Figure 8.3 (d) Neighbour points localization

8.1.1.4 Centreline reconstruction

Figure 8.4 describes three steps of centreline reconstruction: navigation, deletion and connection. By using these steps, a reconstructed centreline can be established and saved in a centreline file. Figure (a) describes the original centreline of a real patient vasculature, and figure (b) shows the end points selected for navigating reconstruction process. The speed of running localization (i.e. selection) algorithm depends on how many candidate points for selection. For example, 236 points within 1s is enough. Figure (c) shows the updated centreline after reconstruction, the waiting time for deletion depends on the candidate points for deletion, nearly 5s in this case. Normally, using end points as candidate points is quite efficient; however, using all points or control points needs more waiting time. The connection process is quite fast (0.1s) since only two points are selected as candidate points if running manually in this case. The updated centreline (Figure 8.5) can be used for further calculation to obtain more information as described in chapter 4.

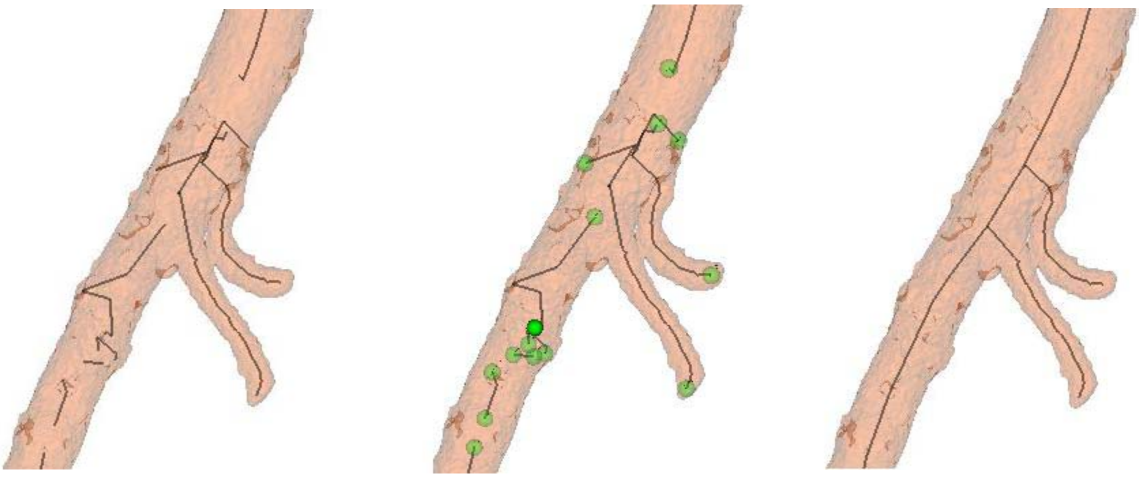


Figure 8.4 (a) Original centreline of a real patient vasculature with five branches

(b) Using End points for navigating reconstruction process.

(c) Updated Centreline after reconstruction including deletion and connection processes.

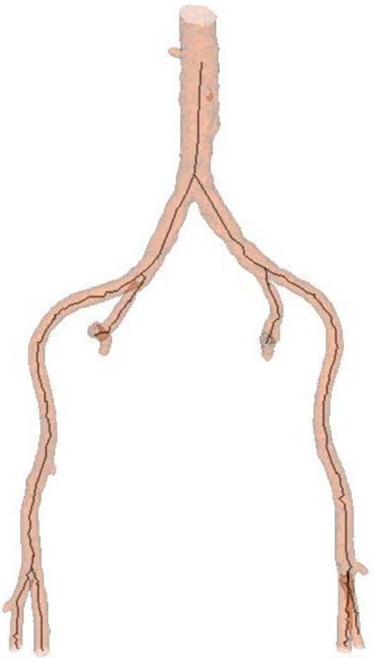


Figure 8.5 Updated centreline of a patient vasculature generated by the centreline reconstruction tool as described in chapter 4.

552 unnecessary points are filtered compared to the original centreline, and 273 unnecessary lines are filtered within 5 minutes including 10s for running automatic algorithms, about 60s for repeat localization navigation points, and nearly 4 minutes for using manual processing (deletion and connection).

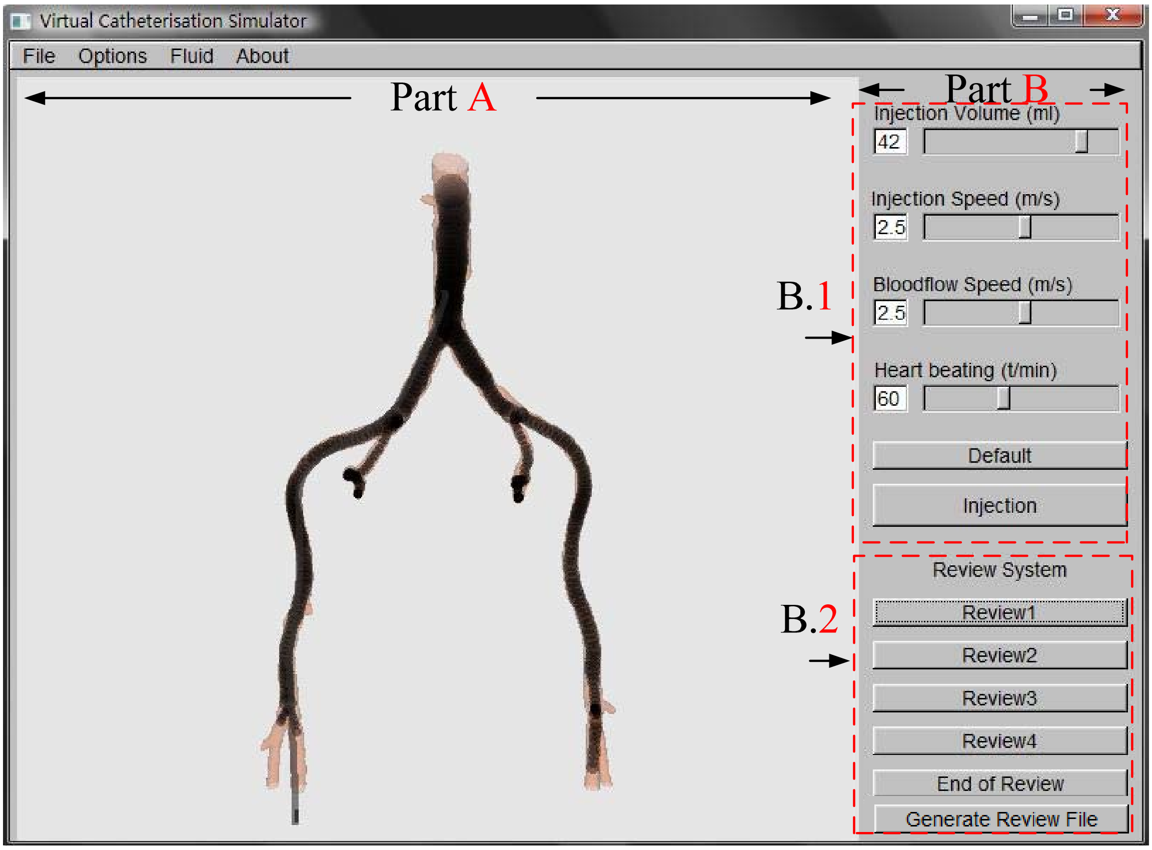
11 branch lines are obtained just like the vascular has 11 branches, and are written in the final centreline file. Each branch line begins with a bifurcation point, and ends with either a bifurcation point or an end point. The previous effect is accomplished by using the merging algorithm.

8.1.2 Blood flow and contrast medium propagation simulation

In the previous section, a reconstructed centreline is generated, and important key points are also localized. They are all well organized in the files and then loaded by the virtual catheterisation simulator framework. In chapter 5 and chapter 6, blood flow and contrast medium simulation are implemented, and the result of simulation is presented in this section. Section 8.1.2.1 describes the injection control panel and graphic view, and section 8.1.2.2 describes the injection using the VSP haptic device, section 8.1.2.3 shows the whole process of contrast medium propagation under the effect of blood flow, and section 8.1.2.4 presents the simple review system.

8.1.2.1 Control panel and graphic view

Figure 8.6 illustrates the blood flow and contrast medium propagation simulation environment in the Virtual Catheterisation Simulator (VCSim), which includes two parts.



Injection Volume (ml) 42

Injection Speed (m/s) 2.5

Bloodflow Speed (m/s) 2.5

Heart beating (t/min) 60

Figure 8.6 Virtual Catheterisation Simulator – Blood flow and contrast medium propagation simulation

Part A is the graphic view. It provides a virtual environment for clinicians to see their injection of the contrast medium during the angiogram. Besides, users can change the view point by using the left mouse button (LBM), or change the viewing distance by using the right mouse button (RBM).

Part B is the control panel. It combines two sub parts. B.1 is the control area for changing the injection volume and speed, as well as configuring the max blood flow speed around the injection position, and modifying the heart beating times per minutes to simulate various patient physiologies. B.1 also includes an injection button providing an alternative to start the contrast injection. B.2 is a review system control part, four stages during the whole injection process can be reviewed and evaluated, besides, the review file can also be generated as described in section 7.4.

8.1.2.2 Contrast medium propagation with blood flow

Figure 8.7 illustrates a contrast medium injection case in a real patient vascular model. The heart beating of the patient is set as 60 times per minute, and injection speed is 2.5m/s, injection volume is 3ml, and maximum blood flow velocity varies across branches as implied by Poisseuille law.

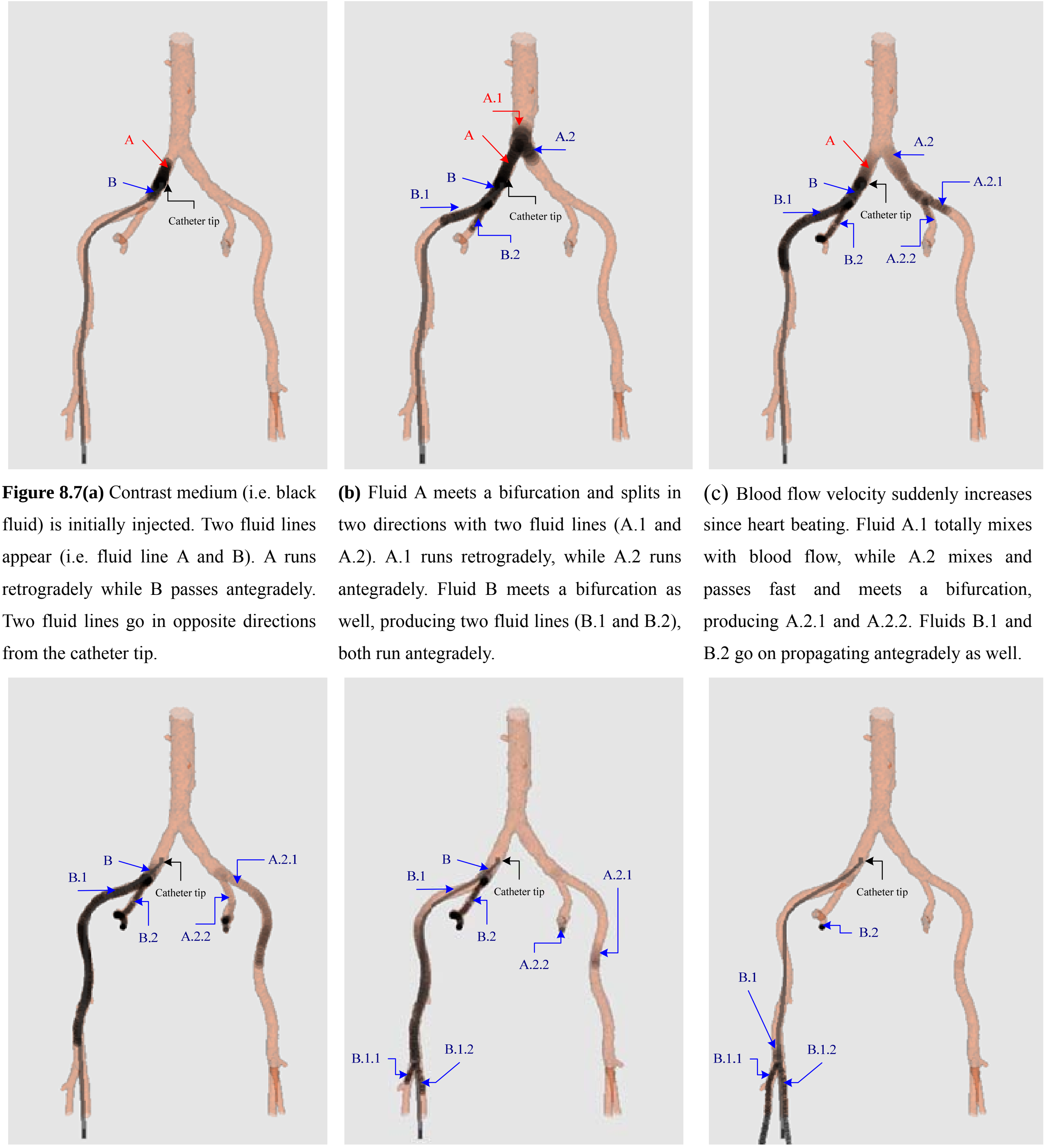


Figure 8.7(a) Contrast medium (i.e. black fluid) is initially injected. Two fluid lines appear (i.e. fluid line A and B). A runs retrogradely while B passes antegradely. Two fluid lines go in opposite directions from the catheter tip.

(b) Fluid A meets a bifurcation and splits in two directions with two fluid lines (A.1 and A.2). A.1 runs retrogradely, while A.2 runs antegradely. Fluid B meets a bifurcation as well, producing two fluid lines (B.1 and B.2), both run antegradely.

(c) Blood flow velocity suddenly increases since heart beating. Fluid A.1 totally mixes with blood flow, while A.2 mixes and passes fast and meets a bifurcation, producing A.2.1 and A.2.2. Fluids B.1 and B.2 go on propagating antegradely as well.

(a) As time goes by, fluids A and A.2 totally mix with the blood flow and also joins another fluid. A.2.1 and A.2.2 mix fast since the CM volume of A.2 is not much. The fluid B and A joins B.1 and B.2 and propagates antegradly.

(a) Blood flow velocity increase suddenly again, fluid B, B.1, A.2.2 and A.2.1 mixes and propagates quickly. B.2 still appears since cross-section is small and the vasculature is complex. B.1 meets a bifurcation and B.1.1 and B.1.2 appear.

(a) As time goes by, fluids B, A.2.2 and A.2.1 mix and join another fluid. B.1 is nearly mixed in his branch, while B.2 still exists in a small area of his branch. The residual CM in B.1.1/2 will pass and mix antegradely until the bottom of the vessel.

8.1.2.3 Review system

With the review system, four stages of the whole injection process can be recorded and display as needed. It can be used as a roadmap to guide during the catheter and guidewire navigation. The five snapshots presented in figure 8.8 show the catheter in a vasculature and the injection of contrast medium at four time points. Even if at some time point contrast medium is mixed, it still can be displayed regarding to the previously recorded information. Besides, their dynamic states such as velocity and blood flow situation are all recorded as described in section 7.4.

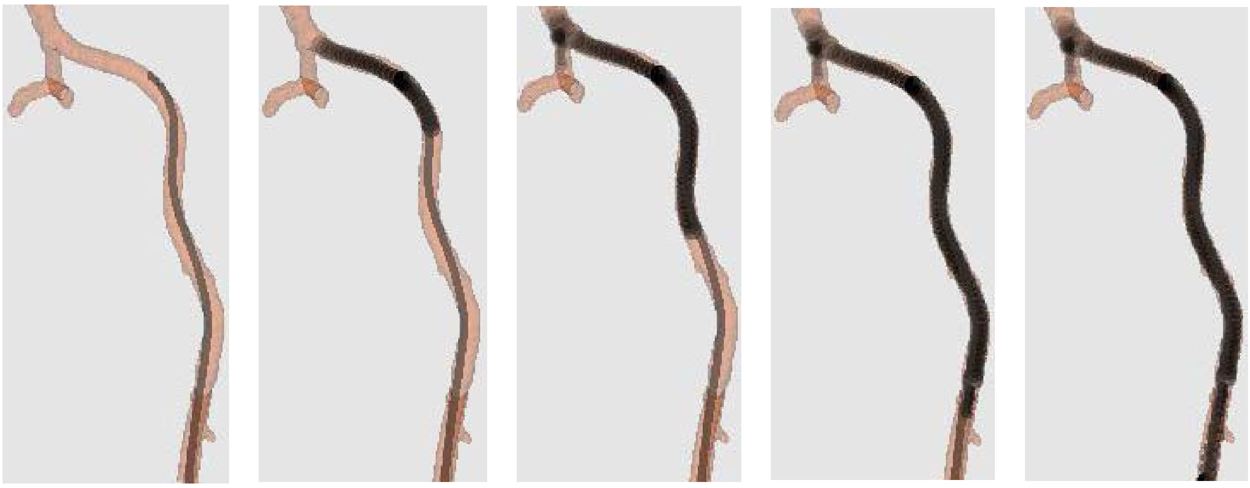


Figure 8.8(a) Start **(b)** Review 1. **(b)** Review 2. **(b)** Review 3. **(b)** Review 4.

8.2 Application to different patient-specific vasculature cases

One of the advantages of our simulator is that it supports various patient-specific vasculatures, thus the blood flow and contrast medium propagation can be applied to new virtual patients. So far, 23 real vasculatures have been integrated in our simulator from 22 CTA scans and one MRA scan (mean age 73.21 years, with a standard deviation of 10.06, range 52 to 89; 14 males, 9 females). In each case, the vasculature covers the same arteries: the aorta, the renal arteries, the iliac and femoral arteries and the beginning of the neck vessels [10]. From these patients, three are selected for evaluation of the centreline reconstruction tool, as well as the virtual contrast medium injection environment. The virtual vascular surfaces are generated as the forms of a large number of triangles and smoothed from the 3D vasculature tool. Centrelines are also generated from the datasets using the centreline reconstruction tool. Centreline points (CLP), branches, bifurcations, control points (CL), end points (EP), feature neighbour points (FNP), approximated total time (TT) and MT (manual interaction time) are all obtained and their numbers are summarized for each patient in **table 8.1**.

Patient No.	Triangles	CLP	Branches	Bifurcations	CP	EP	FNP	TT	MIT
RLH002	54500	239	11	5	77	7	15	3min	2.5min
RLH013	167464	194	11	5	53	7	15	5min	4min
RLH007	219904	236	11	5	73	7	15	5min	4min

Table 8.1 Models and Centreline information of tree Patients vasculatures

The demonstration presented earlier in this chapter used the patient dataset RHL002. Figure 8.9 illustrates both the processing of centreline of the vasculature of patients RHL013 and RHL007 using CRTool, and the contrast medium (CM) propagation in those vessel models. The videos of CM propagation in these three patient vasculatures are recorded using Magic Img2Ani Software. Later, more complex patient-specific vascular models could also be integrated in the simulator.

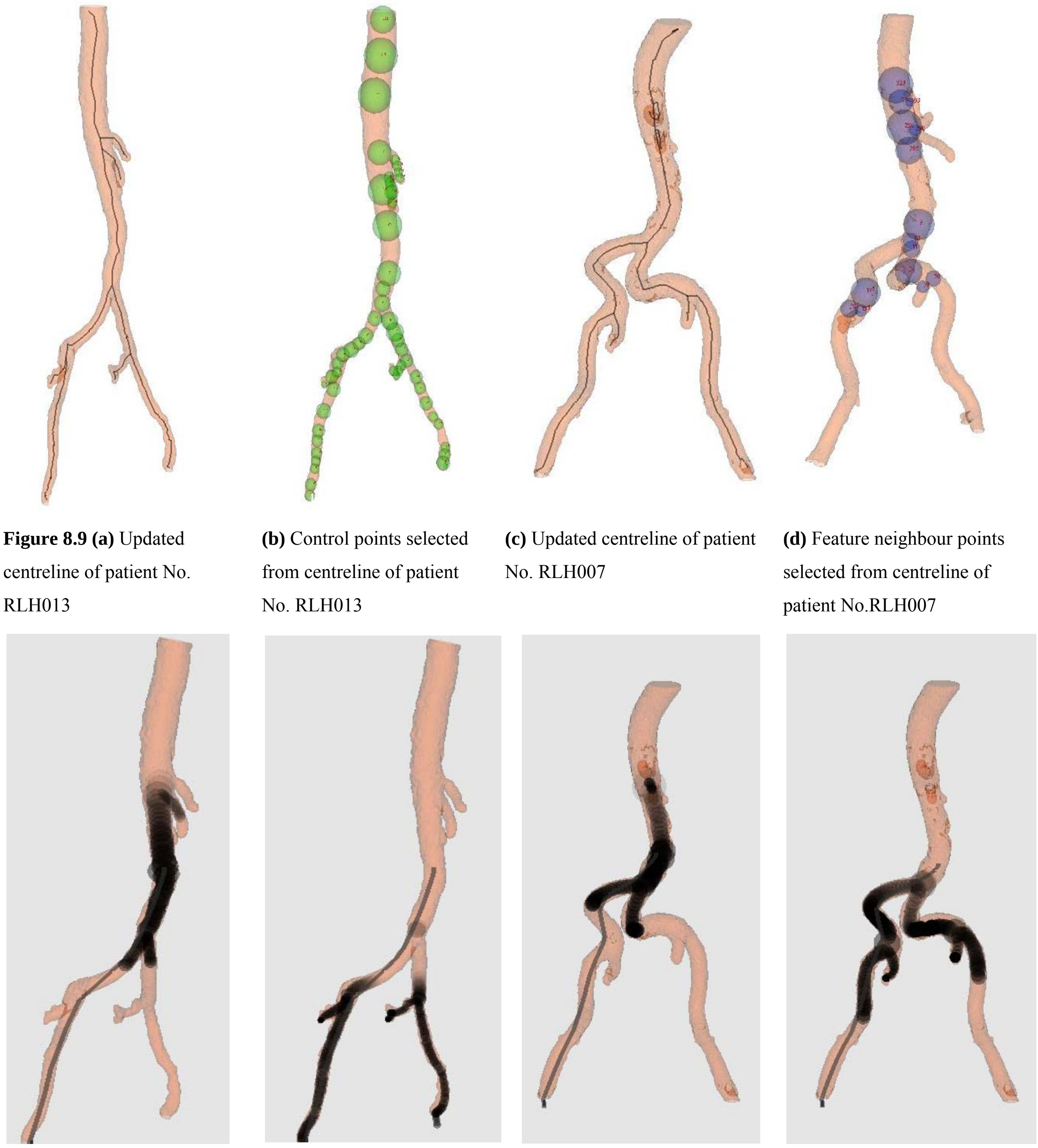


Figure 8.9 (a) Updated centreline of patient No. RLH013

(b) Control points selected from centreline of patient No. RLH013

(c) Updated centreline of patient No. RLH007

(d) Feature neighbour points selected from centreline of patient No.RLH007

(e) CM propagating 1s after injection in the patient No. RLH013

(f) CM propagating 2s after injection in the patient No. RLH013

(g) CM propagating 1s after injection in the patient No. RLH007

(g) CM propagating 2s after injection in the patient No. RLH007

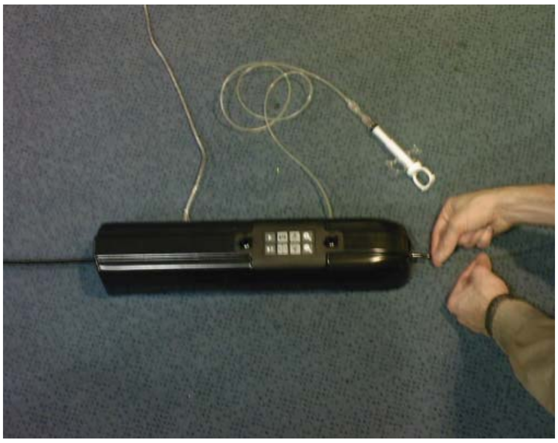
Table 8.2 describes the analysis of the contrast medium simulation, where patient No includes the three patients' datasets, triangles specify the number of triangle of the vascular model. CAP (catheter particles) means the length of catheter from the original position (measure the number of particles), which is the same for all the tests, and position describes in which branch the injection happens. Patient No, Triangles, CAP and Position describe the basic environment for the evaluation. Then, IV (injection volume), IS (injection speed), BFS (blood flow speed) and HB (heart beats) are chosen based on experience for a standard patient for simulating various injection parameters. Finally, CMP (contrast medium particles) and Average FPS (Frames per second obtained from FRAPS (http://www.fraps.com/)) shows the speed for rendering. The tests are performed under a Dell Inspiron I1420 laptop with the following characteristics: Intel(R) Core(TM) 2 Duo CPU T1800 at 2.10GHZ and 2GB RAM, using Windows VistaTM Home Basic operation system. The average FPS for all fifteen tests is 20.8 with standard deviation of 6.3, which is about 7 times faster than the real guided medical imaging for contrast medium propagation (3 FPS). However, the rendering speed is highly dependent on how many contrast medium particles are rendered, thus it needs to be improved later. Among other, the blood flow speed is recorded at the position of injection and at the time point when the contrast medium is injected.

Patient No	Triangles	CAP	Position	IV(ml)	IS(m/s)	BFS(m/s)	HB(t/m)	CMP	Ave FPS	CPU(GHZ)	RAM(GB)
RHL002	54500	19	Femoral	2	2.5	2.5	50	201	31	2.10	2
RHL002	54500	67	Iliac	1	2.5	2.38	60	41	31	2.10	2
RHL002	54500	92	Iliac	3	2.5	3.32	60	198	19	2.10	2
RHL002	54500	118	Iliac	3	5	2.42	90	291	16	2.10	2
RHL002	54500	136	Aorta	1	3.8	3.67	90	202	21	2.10	2
RHL007	219904	33	Iliac	3	4	0.15	60	101	21	2.10	2
RHL007	219904	77	Iliac	2.4	5	2.27	70	101	26	2.10	2
RHL007	219904	127	Iliac	4.2	5	0	50	393	19	2.10	2
RHL007	219904	165	Aorta	2.4	4	2.3	90	522	12	2.10	2
RHL007	219904	165	Aorta	1	4	0.92	90	62	18	2.10	2
RHL013	167464	25	Iliac	3	2.5	1.2	60	100	28	2.10	2
RHL013	167464	55	Iliac	3	4	0	50	206	26	2.10	2
RHL013	167464	94	Iliac	1.8	5	2.3	100	417	16	2.10	2
RHL013	167464	124	Aorta	3	4.5	2.27	60	1281	12	2.10	2
RHL013	167464	195	Aorta	1.6	1.2	1.3	90	144	16	2.10	2

Table 8.2 Analysis of the contrast medium simulation

8.3 Contrast medium injection using VSP

The trainees can use the VSP haptic device to interact with the computer, and receive force feedback from their manipulations in the vasculature. Firstly, they can insert a guidewire into the vessel (mainly at the femoral artery) after operating the Seldinger technique (consisting of inserting a needle in the artery). However, in the real clinical environment, users can see the 3D vasculature only under fluoroscopy and while contrast agent is injected to highlight the vessels. Thus, trainees should use the virtual contrast medium injection to obtain the roadmap and guide the next steps of the angiography. They can use the VSP syringe, specifying parameters such as blood flow speed on the control panel, and using the syringe just like in the real clinical environment. After the lifetime of one injection ends, they can use the review system to obtain the roadmap, and go on operating on the patient (Figure 8.10).



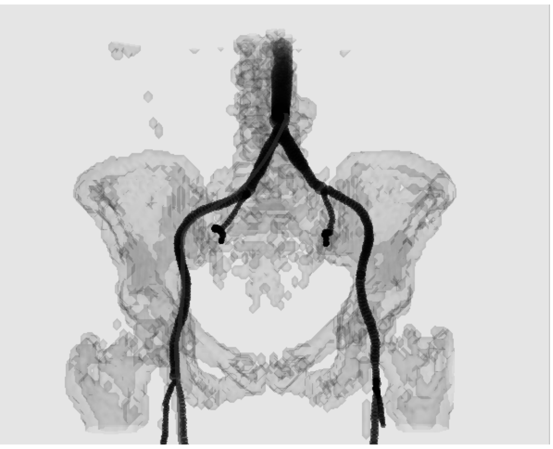


Figure 8.10 Using the VSP haptic device for catheterization and angriography (left), and constructing the roadmap thanks to the contrast medium injection under fluoroscopy environment (right).

8.4 Feedback from experts

The recorded videos of blood flow and contrast medium propagation simulation in three patients' vasculature were demonstrated during the quarterly project meeting of the CRaIVE (Collaboration in Radiological Interventional Virtual Environments) consortium. Scientists from both computing and radiological interventional fields discussed on the demonstration and gave evaluations for the simulation, providing suggestions on the virtual fluid propagation speed and duration of injection. Overall, they thought that the simulation is very realistic and corresponds to the real contrast medium propagation process. They thought that the simulation could be used in the virtual interventional radiology environment to model angiography during the diagnosis process.

After the meeting, Dr. Derek A. Gould, Consultant Radiologist from Royal Liverpool University Hospital and Chair of the Collaboration in Radiological Interventional Virtual Environments (CRaIVE) consortium, gave specific comments on the contrast medium simulation. For him, the states of pause and acceleration of contrast medium during propagation realistically translate the influence of a real beating heart, and the average of velocity of contrast medium is also very realistic, especially for the young patients with high blood flow velocity. The contrast medium injected would go retrogradely not much higher in the aorta, and at the same time, blood mixed contrast medium would pass anterogradely with the rhythm of beating heart and changeable blood flow velocity. The previous phenomenon is also well simulated and performed realistically. Further, the route of contrast medium propagation from the beginning until the end is accurate as well, like the real phenomenon: the fluid would go not much faded until the bottom of the vasculature. Then, he provided suggestions for improvements of the contrast medium velocity when meeting vasculature situations of old patients. Moreover, he wondered if the extension to various blood flow states such as turbulence would be useful and would improve the realism. He also suggested further integration of the CM simulation in the main project of real-time Seldinger technique simulator.

9. Conclusion and future work

This chapter concludes the whole project, summarising all achievements, discussing its results, and giving recommendations for future work. Mathematical background and abbreviations are outlined in Appendix A and B, respectively. In the end of thesis, a user's guide is included for guiding the use of the reconstruction tool and the injection application.

9.1 Achievements

This project provides an innovative way for blood flow and contrast medium (CM) simulation, the achievements include:

(1) Extensive survey of training for interventional radiology procedures and catheterisation techniques, and also discuss the advantages and disadvantages of various ways for contrast medium fluid simulation. Besides, basic fluid mechanics are introduced and the deduction from Navier-Stoke equation to Poisseuille Law is excerpted.

(2) Create an innovative framework for blood flow and contrast medium simulation regarding to the prior information obtained from centreline points and cross-section size, and present realistic validated results of this framework.

(3) Create a centreline reconstruction tool in order to obtain a good quality vascular centreline, as well as get accurate cross-section information of models. Besides, using this tool enables users to get more specific information such as bifurcation points, control points, end points and feature neighbour points for contrast medium propagation.

(4) Create a new approach for blood flow velocity calculation. Maximum blood flow velocity is calculated using the CRTool and attached as a scalar value to the corresponding control points. The dynamic blood flow with the effect of beating heart is obtained in relation to the maximum value and the heart beating model.

(5) Create the isolated heart beating model using a sine function for various patient cases by changing the amplitude of the function. From the model, dynamic velocities of the blood flow can be read after input of the maximum blood flow value. Besides, the acceleration of blood flow and mixing parameters can also be obtained.

(6) Create a series of models and systems for achieving the realistic propagation of contrast medium with the effect of blood flow, including fading system, life control system, velocity changing system, adaptive radius changing system, and so forth.

(7) Design a new two particle line model for initialization of CM injection, as well as CM propagation in different directions through various branches thanks to bifurcation detection and response. Theoretically, these models can deal with complex vasculature models with even 'infinite' bifurcations.

(8) Add the protocol of review system, and the whole injection process can be recorded and replayed as a roadmap. A review file can be generated and stored as a basis for further evaluation.

(9) Combine the VSP haptic device with the virtual contrast medium environment so that users could use the VSP syringe for CM injection. Theoretically, users can inject even 'infinite' times in various areas of the vasculature model.

9.2 Future work

Based on the created framework for CM simulation, possible future work mainly concentrates on four parts:

(1) Update centreline reconstruction tools for optimizing filtering and automatic connection speed. More automatic algorithms rather than manual construction methods should be designed and evaluated. Besides, some patient-specific cases with serious aneurysm with sharp changing rate of cross-section size should be tested in the CM simulation framework to see how the simulator deals with those extreme cases. If problems occur, the configuration file might need to be re-designed to meet this new challenge by storing more centreline information. A user-friendly interface of the CRTool needs to be considered. For example, more flexible and simple interaction methods using mouse controls rather than inputting indexes of points should be designed and implemented.

(2) Enhance the realism of the contrast medium simulation based on published medical papers and suggestions from clinicians. Such as more effective physics modelling for contrast medium velocity and mixing systems. Further, blood flow models should also be updated by using more accurate and real models, integrating real cardiogram for the replacement of the sine beating heart model, and using more complex mathematic simulation of heart model as proposed in [60].

(3) Optimize the real-time speed of contrast medium simulation based on algorithm optimization and GPU acceleration. Although the fluid propagates in real-time so far, this cannot be guaranteed if the particle numbers become too large to be rendered quickly. In consequence, more pre-computing process is an idea for accelerating the speed while not reducing the realistic effects. Moreover, more efficient data structures should be used for replacing existing ones to improve the rendering algorithms. Furthermore, GPUs could be used to compute information on large particle lines as they mostly repeat the same simulation steps and would allow a substantial acceleration.

(4) Translating the blood flow and contrast medium simulation to more patients. One of our project collaborators, Dr. C.J. Hughes, developed a Macroscopic blood flow visualization using boids. It would be interesting to combine our projects together to achieve a more realistic interaction between blood flow and contrast medium, as well as modelling different complex states of blood flow [67].

Bibliography

[1] Luboz V., Blazewski R., Gould D., Bello F. Real-time guidewire simulation in complex vascular models. The Visual Computer. Springer, Berlin/Heidelberg.ISSN 0178-2789 (Print) 1432 – 2315. (2009)

[2] Imperial College Course *Interactive Graphics* Lecture Notes.

[3] Darzi A., Munz Y. The impact of minimally invasive surgical techniques. Annu. Rev. Med. 55:223-237. doi: 10.1146/annurev.med.55.091902.105248. (2004)

[4] John J. Regan. Robotics and Computers in Minimally Invasive Spine Surgery. [Accessed: 19 March 2009]. http://www.spineuniverse.com/displayarticle.php/article1511.html.

[5] isnake. http://www1.imperial.ac.uk/medicine/news/isnake/.

[6] David Sutton. Radiology and imaging for medical students (Sixth edition). Edinburgh London Madrid Melbourne New York and Tokyo 1994. ISBN 0-443-04883-5.

[7] Adel B., Laurent D., Ibrahima M., Multiobjective optimization of a stent in a fluid-structure context. Proceedings of the 2008 GECCO conference companion on Genetic and evolutionary computation. (2008).

[8] VascularWeb. Angioplasty and Stenting.
http://www.vascularweb.org/patients/NorthPoint/ Angioplasty_and_Stenting.html.

[9] Wu, X., Allard, J., Cotin, S., Real-Time Modeling of vascular flow for angiography simulation. In MICCAI 2007, Part I, LNCS 4791, pp.557-565, 2007.

[10] Luboz V., Hughes C., Gould D., John N. and Bello F., Real-time Seldinger technique simulation in complex vascular models. International Journal of Computer Assisted Radiology and Surgery. 1861-6410. (2009)

[11] Chui, C.K., Li, Z., Anderson, J.H., Murphy, K., Venbrux, A., Ma, X., Wang, Z., Gailloud, P., Cai, Y., Wang, Y., Nowinski, W.L.: Training and pretreatment planning of interventional neuroradiology procedures – initial clinical validation. Medicine Meets Virtual Real. 85, 96–102 (2002).

[12] Luboz, V., Lai, J., Blazewski, J., Gould, D., Bello, F.: A virtual environment for core skills training in vascular interventional radiology. Proceedings of the 4th International Symposium on Biomedical Simulation, vol.5104, pp.215–220. (2008)

[13] Liu W. Sewell C. Blevins N. Salisbury K. Bodin K. Hjelte N. Representing Fluid with Smoothed Particle Hydrodynamics in a Cranial Base Simulator. Stud Health Technol Inform.132,257-9. (2008)

[14] Schroeder W. Martin K. Lorensen B. The Visualization Toolkit 2nd Edition. Prentice Hall PTR. (1998)

[15] Jou L. Rapp J. Saloner D. Transport of contrast agents in contrast-enhanced magnetic resonance angiography. Magnetic Resonance Imaging 22. 495–504. (2004)

[16] Horkaew, P. Merrifield, R. Yang, G.Z. Building optimal statistical deformable surface models Information Technology Applications in Biomedicine, 2003. 4th International IEEE EMBS Special Topic Conference on. 215- 218. (2003)

[17] Koch, R.M., Gross, M. H., Carls, F. R., Bren, D. F., Fankhauser, G., and Parish, Y. I. H., Simulating facial surgery using finite element models. ACM SIGGRAPH 421–428, 1996.

[18] Xia, J., James, X., Horace, H. S. I., Nabil, S., Helena, T. F. W., Jaime, G., Dongfeng, W., Richie, W. K. Y., Christy, S. B. K., and Henk, T., Three- dimentional virtual-reality surgical planning and soft-tissue prediction for orthognathic surgery. IEEE Trans. Inform. Technol. Biomed. 5(2):97–107, 2001.

[19] Preim, B., Spindler, W., Oldhafer, K. J., and Peitgen, H. O., 3DInteraction techniques for planning of oncologic soft tissue operations. Proc. Graph. Interface. (2001)

[20] M Nakao.,T Kuroda.,H Oyama.,G Sakaguchi.,M Komeda. Physics-Based Simulation of Surgical Fields for Preoperative Strategic Planning. Journal of Medical Systems.Volume 30. Issue 5. 371 - 380.

(2006)

[21] Poiseuille Law. http://en.wikipedia.org/wiki/Hagen%E2%80%93Poiseuille_equation.

[22]Müller M., Charypar D. Gross M. Partial-Based Fluid Simulation for Interactive Applications. Eurographics/SIGGRAPH Symposium on Computer Animation. (2003)

[23] Concept House website. http://www.concepthouse.com/. [Accessed: 17 March 2009]

[24] W. T. Reeves. Particle systems . a technique for modeling a class of fuzzy objects. ACM Transactions on Graphics 2(2), pages 91.108. (1983)

[25] Bridson R. Muller-Fischer M. FLUID SIMULATION SIGGRAPH 2007 Course Notes (2007)

[26] Gridfluid— 3D Grid Based Fluid Simulator.(2008)

[27] Jou L. Rapp J. Saloner D. Transport of contrast agents in contrast-enhanced magnetic resonance angiography. Magnetic Resonance Imaging 22. 495–504. (2004)

[28] Maki J, Prince M, Londy F, Chenevert T. The effects of time varying intravascular signal intensity and k-space acquisition order on three dimensional MR angiography image quality. J Magn Reson Imaging: 6:642–51. (1996)

[29] Wu E., Zhu H., Liu X., Liu Y. Simulation and interaction of fluid dynamics. Visual Computing 299-308. (2007)

[30] Foster, N., Metaxas, D.: Realistic animation of liquids. Graph. Models Image Process.58 (5), 471–483 (1996)

[31] Stam, J.: Stable fluids. In: Proceedings of SIGGRAPH 1999, pp. 121–128 (1999)

[32] Muller M. Stam J. James D. Thurey N. Real-time physics class notes. SIGGRAPH 2008 .(2008)

[33] A. Fournier and W. T. Reeves. A simple model of ocean waves. In Proc. SIGGRAPH, pages 75–84. (1986)

[34] D. Hinsinger, F. Neyret, and M.P. Cani. Interactive animation of ocean waves. In Proc. ACM SIGGRAPH/Eurographics Symp. Comp. Anim., pages 161–166. (2002)

[35] Long B. Reinhard E. Real-Time Fluid Simulation using Discrete Sine/CosineTransforms. Symposium on Interactive 3D Graphics. Proceedings of the 2009symposium on Interactive 3D graphics and game. Pages 99-106. (2009)

[36] N. Foster and D. Metaxas. Modeling the Motion of a Hot, Turbulent Gas. In Computer Graphics Proceedings, Annual Conference Series, 1997, pages 181–188, August 1997.

[37] Francis J.Percutaneous Transluminal Coronary Angioplasty. http://www.cardiophile.com/percutaneous-transluminal-coronary-angioplasty-pta. [Accessed: 13 March 2009]

[38] Medscape Website. http://www.medscape.com/viewarticle/437305_4. [Accessed: 20 March 2009]

[39] Wiki. http://en.wikipedia.org/wiki/Marching_cubes. [Accessed: 22 March 2009]

[40] Daniel Ryan King, Building of three-dimensional vascular models for interactive simulation in Interventional Radiogy, Project report of Imperial College, 2008.

[41] Patentgenius ebsite. http://www.patentgenius.com/patent/7465434.html. [Accessed: 5 March 2009]

[42] Wiki.http://en.wikipedia.org/wiki/Graphics_processing_unit. [Accessed: 23 March 2009]

[43] Meier U. Lopez O. Monserrat C. Juan M.C. Alcaniz M. Real-time deformation survey 2005. Comp uter Methods and Programs in Biomedicine (2005) 77, 183—197(2005)

[44] GIBSON, S. 1997. 3D ChainMail: A Fast Algorithm for Deforming Volumetric Objects.Proceeding of 1997 Symposium on interactive 3D Graphics.149-154. (1997)

[45] Clavet S. Beaudoin P. Poulin P. Particle-based Viscoelastic Fluid Simulation. Eurographics/ACM SIGGRAPH Symposium on Computer Animation. (2005)

[46] wiki Seldinger. http://en.wikipedia.org/wiki/Seldinger_technique

[47] Healthwise. http://myhealth.ucsd.edu/library/healthguide/en-us/support/topic.asp?hwid=zm2633.

[48] http://jkms.kams.or.kr/fulltext/html/jkms21950h.html. Cavernous Angioma in the Falx Cerebri: A

Case Report

[49] Interventional radiology. http://en.wikipedia.org/wiki/Interventional_Radiology. [Accessed:20 March 2008].

[50] J. P.Morris. Simulating surface tension with smoothed particle hydrodynamics. International Journal for Numerical Methods in Fluids, 33(3):333.353, 2000.

[51] A Hybrid 1D and 3D Approach to Hemodynamics Modelling for a Patient-Specific Cerebral Vasculature and Aneurysm Harvey Ho1, Gregory Sands1, Holger Schmid2, Kumar Mithraratne1, Gordon Mallinson3, and Peter Hunter1.

[52] Imperial College Lecture 06 advanced graphics lecture notes.

[53] Horkaew, P. Merrifield, R. Yang, G.Z. Building optimal statistical deformable surface models Information Technology Applications in Biomedicine, 2003. 4th International IEEE EMBS Special Topic Conference on. 215- 218. (2003)

[54]Rafal Blazewski. Virtual Catheterisation Simulator. Imperial College Postgraduate Dissertation. (2008)

[55]Dr. Vincent Luboz personal website. http://www1.imperial.ac.uk/medicine/people/v.luboz/. [Accessed: 20 March 2009]

[56] Duratti L.,Wang F., Samur E., Bleuler H., A Real-time Simulator for Interventional Radiology. 2008.

[57] Navier-stockes to Poiseuille Law.
http://en.wikipedia.org/wiki/Hagen%E2%80%93Poiseuille_flow_from_the_Navier%E2%80%93Stokes_equations . derive Poiseuille from navier law.

[58] Boundary condition. http://en.wikipedia.org/wiki/Boundary_condition.

[59] Dr songyi's home page . http://www.comp.leeds.ac.uk/yisong/research_current.html .

[60] C.J. Hughes, S.R. Pop, N.W. John, "Macroscopic blood flow visualization using boids", 23rd International Congress of CARS - Computer Assisted Radiology and Surgery, Berlin, Germany, June 2009

[61] Kitware, Inc., "VTK Home Page," The Visualization Toolkit. [Accessed: 20 March 2008]. Available: http://www.vtk.org/.

[62] Xiaoyuan Zhou. Virtual Catheterisation Simulator-Vessel deformations. Imperial College Postgraduate dissertation. (2009)

[63] Bill Spitzak et al., "Fast Light Toolkit (FLTK)," FTLK: Fast light toolkit.
Available: http://www.fltk.org/. [Accessed: 20 March 2009].

[64] Kitware, Inc., "CMake Cross Platform Make," CMake. Available: http://www.cmake.org/. [Accessed: 20 March 2009].

[65] Kitware, Inc., "VTK Home Page," The Visualization Toolkit. Available:http://www.vtk.org/. [Accessed: 20 March 2008].

Appendix

A. Mathematical background

A. 1 Gradient

The gradient simply takes all the spatial partial derivatives of the function, returning a vector. In two dimensions:

$$\nabla\mathrm{f}(\mathrm{x},\mathrm{y}) = \left(\frac{\partial \mathrm{f}}{\partial \mathrm{x}},\frac{\partial \mathrm{f}}{\partial \mathrm{y}}\right) (\mathbf{A.1})$$

And in three dimensions:

$$\nabla\mathrm{f}(\mathrm{x},\mathrm{y},\mathrm{z}) = \left(\frac{\partial \mathrm{f}}{\partial \mathrm{x}},\frac{\partial \mathrm{f}}{\partial \mathrm{y}},\frac{\partial \mathrm{f}}{\partial \mathrm{z}}\right) (\mathbf{A.2})$$

It can sometimes be helpful to think of the gradient operator as a symbolic vector, e.g. in three dimensions:

$$\nabla = \left(\frac{\partial}{\partial \mathrm{x}},\frac{\partial}{\partial \mathrm{y}},\frac{\partial}{\partial \mathrm{z}}\right) (\mathbf{A.3})$$

The gradient is often used to approximate a function locally:

$$\mathrm{f}(\mathrm{x}+\ \Delta\mathrm{x})\ \approx \mathrm{f}(\mathrm{x}) + \nabla\ \mathrm{f}(\mathrm{x}) \cdot \Delta\mathrm{x}\ (\mathbf{A.4})$$

In a related vein we can evaluate the “directional derivative” of the function that is how fast it is changing when looking along a particular vector direction using the gradient. For example, if the direction is n:

$$\frac{\partial \mathrm{f}}{\partial \mathrm{x}} = \nabla\mathrm{f} \cdot \mathrm{n}\ (\mathbf{A.5})$$

Occasionally we will take the gradient of a vector valued function, which results in a matrix (sometimes called the “Jacobian”). For example, in three dimensions:

$$\nabla\mathrm{f} = \ \nabla(\mathrm{f},\mathrm{g},\mathrm{h}) = \begin{pmatrix} \frac{\partial \mathrm{f}}{\partial \mathrm{x}} & \frac{\partial \mathrm{f}}{\partial \mathrm{x}} & \frac{\partial \mathrm{f}}{\partial \mathrm{x}} \\ \frac{\partial \mathrm{f}}{\partial \mathrm{x}} & \frac{\partial \mathrm{f}}{\partial \mathrm{x}} & \frac{\partial \mathrm{f}}{\partial \mathrm{x}} \\ \frac{\partial \mathrm{f}}{\partial \mathrm{x}} & \frac{\partial \mathrm{f}}{\partial \mathrm{x}} & \frac{\partial \mathrm{f}}{\partial \mathrm{x}} \end{pmatrix} (\mathbf{A.6})$$

Note that each row is the gradient of one component of the function. The matrix-vector product is just computing the dot-product of each row of the matrix with the vector, and so each row should be a gradient of the function:

$$\nabla(\mathrm{f,g,h}) = \begin{matrix}\nabla \mathrm{f} \\ \nabla \mathrm{g} \\ \nabla \mathrm{h}\end{matrix} \ (\mathbf{A.7})$$

A. 2 Laplacian

The Laplacian is usually formed as the divergence of the gradient. Sometimes it is written as ∇^2 or Δ, but since these symbols are occasionally used for other purposes, I will stick to writing it as $\nabla \cdot \nabla$. In two dimensions:

$$\nabla \cdot \nabla \mathrm{f} = \frac{\partial^2 \mathrm{f}}{\partial \mathrm{x}^2} + \frac{\partial^2 \mathrm{f}}{\partial \mathrm{y}^2} \ (\mathbf{A.8})$$

And in three dimensions:

$$\nabla \cdot \nabla \mathrm{f} = \frac{\partial^2 \mathrm{f}}{\partial \mathrm{x}^2} + \frac{\partial^2 \mathrm{f}}{\partial \mathrm{y}^2} + \frac{\partial^2 \mathrm{f}}{\partial \mathrm{z}^2} \ (\mathbf{A.9})$$

The Laplacian can also be applied to vector or even matrix fields, and the result is simply the Laplacian of each component.

A. 3 Taylor expansion

Assuming a function f has at least k smooth derivatives, then

$$\mathrm{f(x + \Delta x) = f(x)} + \frac{\partial \mathrm{f}}{\partial \mathrm{x}}(\mathrm{x})\Delta \mathrm{x} + \frac{1}{2}\frac{\partial^2 \mathrm{f}}{\partial \mathrm{x}^2}(\mathrm{x})\Delta \mathrm{x}^2 + \frac{1}{6}\frac{\partial^3 \mathrm{f}}{\partial \mathrm{x}^3}(\mathrm{x})\Delta \mathrm{x}^3 + \cdots + \frac{1}{(k-1)!}\frac{\partial^{k-1} \mathrm{f}}{\partial \mathrm{x}^{k-1}}(\mathrm{x})\Delta \mathrm{x}^{\mathrm{k}-1} + \mathrm{R_k} \ (\mathbf{A.10})$$

The remainder term $\mathrm{R_k}$ can be expressed in several ways, for example:

$$\mathrm{R_k} = \int_{x}^{x+\Delta x} \frac{1}{k!}\frac{\partial^k f}{\partial x^k}(s) s^{k-1}\, ds \ (\boldsymbol{A.11})$$

$$\mathrm{R_k} = \frac{1}{k!}\frac{\partial^k f}{\partial x^k}(s)\Delta \mathrm{x^k} \text{ for some s} \in [x, x+\Delta x] \ (\mathbf{A.12})$$

$$R_k = O\left(\Delta \mathrm{x^k}\right) \ (\mathbf{A.13})$$

Note that Δx could be negative, in which case the second form of the remainder uses the interval $[x, x+\Delta x]$. We will generally stick to the last form, using the simple $O()$ notation, however, Taylor series is not particularly useful, thus it should be approximated in finite differences in space.

B. Abbreviations

2D	–	2 Dimensional
3D	–	3 Dimensional
AABB	–	Axis Aligned Bounding Box
CT	–	Computer Tomography
OO	–	Object Oriented
VCSim	–	Virtual Catheterisation Simulator
IR	–	Interventional Radiology
VA	–	Virtual reality
LMB	–	Left Mouse Button
RMB	–	Right Mouse Button
IO	–	Input /Output
CMV	–	Contrast Medium Velocity
BV	–	Blood Flow Velocity
CS	–	Cross-section
CM	–	Contrast Medium
BL	–	Blood Flow
CLP	–	Centreline Points
CP	–	Control Points
EP	–	End Points
FNP	–	Feature Neighbour Points
CRTool	–	Centreline Reconstruction Tool
N-S	–	Navier-Stokes
CAP	–	Catheter Particles
IV	–	Injection Volume
IS	–	Injection Speed
BFS	–	Blood Flow Speed
HB	–	Heart Beats
CMP	–	Contrast Medium Particles
FPS	–	Frames Per Second

User-Manual

A. Centreline reconstruction tool

Interface	**Usage**	**Operation**
Translation – X: scroll bar	Translate input units along X axis for the model	I+E/S
Translation – Y: scroll bar	Translating input units along Y axis for the model	I+E/S
Translation – Z: scroll bar	Translating input units along Z axis for the model	I+E /S
Rotation – X: scroll bar	Rotating input units by X axis for the model	I+E /S
Rotation – Y: scroll bar	Rotating input units by Y axis for the model	I+E /S
Rotation – Z: scroll bar	Rotating input units by Z axis for the model	I+E /S
Zero Coordinates	Set all the value to 0 for the Translation and Rotation	C
Load CentreLine	Load the standard centreline file with format .txt into the tool	C
R: input editor	Filter the unnecessary centreline points automatically by the **R**adius threshold	I+E
Filter Unnecessary Lines	Filter the unnecessary centreline lines automatically	C
End Points	End points localization	C
End Points - Del	Delete unnecessary end points/Mark points for connection	I + E
End Points - Branch	Delete the unnecessary lines attached on that end points	C
All Points	All centreline points localization	C
All Points - Del	Delete unnecessary centreline points/Mark points for connection	I + E
All Points - Branch	Delete the unnecessary lines attached on that centreline points	C
Connection	Connect the two centreline points marked before	C
Save	Temporary save function	U
Feature	Feature (i.e. bifurcation) points localization	C
Feature – No:	Select the certain feature point	I+E
Feature – Type:	Confirm the type of bifurcation point (default: three branches attached)	I+E
From:	Select the beginning point for connection, usually it is one of the feature points	I+E
To:	Select the end point for connection, usually it is feature or end point	I+E
Merge	Merge the branch using the begin and end point selected before	C
Save Centreline	Save the centreline to a .txt file	C
Fluid	Feature neighbour points localization	C
Fluid – No:	Select the certain feature neighbour points	I+E
Size	Radius size modification(0-1) for selected centreline points (feature, fluid etc)	I+E
Control Points Localization	Control Points Localization	C
Density:	Select the density value (0-1) of the control points from all the centreline points	I+E
Del:	Delete unnecessary control points	I+E
Control Points Size:	Radius size modification(0-1) for all control points	I+E
Viscosity	Blood flow viscosity value	I+E
Poiseuille Law	Computing maximum blood flow speed on each control point, and then should save a new centreline file	C
Generate Files	Generate centreline configuration file	C
Exit	Exit the centreline reconstruction Tool	

Note: I+E: Input and Enter; C: Click the button; S: Drag the scroll bar; U: Under construction

B. Virtual contrast medium injection environment

The user-manual for VCSim is well documented in [54], and the following table is to describe the user-manual for the virtual contrast medium injection environment newly added.

Interface / Keyboard	**Usage**	**Operation**
Menu - Fluid	Show contrast medium injection control panel	C
Injection Volume (ml)	Input/Select the CM injection volume (default 25 = 3ml)	I+E/S
Injection Speed (m/s)	Input/Select the injection speed value (default 2.5m/s)	I+E /S
Blood flow speed (m/s)	Input/Select the maxi speed value of the blood flow(default 2.5m/s)	I+E /S
Heart beat (t/min)	Input/select how many times the patients' heart beats per minute	I+E /S
Default	Roll back to the default values	C
Injection	Inject the CM	C
Review1	Show the road map of the first stage of injection	C
Review2	Show the road map of the second stage of injection	C
Review3	Show the road map of the third stage of injection	C
Review4	Show the road map of the fourth stage of injection	C
End of Review	Clear the road map	C
Generate review file	Generate the review configuration file	C
Keyboard – space bar	Inject the CM with default values	C

Note: I+E: Input and Enter; C: Click the button; S: Drag the scroll bar; U: Under construction