Surgical Training on the World Wide Web

by

Nuha H. El-Khalili

Submitted in accordance with the requirements
for the degree of Doctor of Philosophy.

The University of Leeds
School of Computer Studies

April 1999

The candidate confirms that the work submitted is her own and the appropriate credit has been given where reference has been made to the work of others.

Abstract

The World Wide Web as a repository of information has had a great influence on our lives. This influence is increasing as the web introduces applications in addition to information. These applications have several advantages, such as world wide accessibility, distance group learning and collaboration. Furthermore, the web encourages training applications since it offers multi-media that can support all stages of training.

On the other hand, the virtual reality technology has been utilised to provide new systematic training methods for surgical procedures. These solutions are usually expensive in terms of cost and computation. In this thesis we propose a novel solution to fulfill the training needs of radiologists performing one type of minimally invasive surgery known as interventional radiology. Our training method combines the capabilities of virtual reality to provide realistic simulation environment together with the web environment to provide platform independent, scalable and accessible system.

In this thesis we analyse this type of surgical procedure in order to deduce the training requirements of such an application. Then, we investigate the possibility of fulfilling these requirements within the server-client architecture of the web environment. We study the degree to which current web technologies- such as Java and VRML- can support the development of a three-dimensional virtual environment with complex interactions. Furthermore, we study the plausibility of providing high computational behaviour modelling training environment on the web by utilising physically-based modelling techniques.

We also discuss the effect of adopting the web environment on fulfilling the virtual reality and training requirements of our system. Finally, we evaluate the resulting system to find out how useful is the proposed solution from the clinical point of view.

Acknowledgements

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

My deepest gratitude goes to my supervisor Dr. Ken Brodlie for his guidance and support.

I would like to acknowledge radiologists and surgeons from Leeds General Infirmary (Dr. Nick Phillips) and St. James's University Hospital (Dr. David Kessel and his colleagues) for their time and medical support that was needed in this work. I also acknowledge Dr. Andy Bulpitt, Dr. Tim David, Dr. Peter Jimack and Dr. Jason Wood for their help during different stages of this work. This extends to the digital media development staff at the University Media Services for their expert assistance in producing the movies for this work.

I would also like to thank colleagues in my department for their technical and moral support: Ying Li, Adriano Lopes, Stuart Lovegrove, Martin Thompson and Rik Wade.

Many thanks also to my friends: Huda Al-fodari, Amal Mansi and Amir Zied for sharing the whole experience with all the ups and downs!

Finally, special thanks goes to my parents and my family whom without their support this PhD would have been mission impossible.

Declarations

Some parts of the work presented in this thesis have been published in the following articles:

Nuha H. El-Khalili, Ken W. Brodlie, "Distributed VR Training System for Endovascular Procedures", *Proceedings of the fourth UK VRSIG Conference*, ed. Richard Bowden, November 1st, Brunel University. (1997) 110–119.

Nuha H. El-Khalili, Ken W. Brodlie, "Architectural Design Issues for Web-Based Virtual Reality Training Systems", *Proceedings of the 1998 International Conference on Web-Based Modeling & Simulation*, ed. Paul Fishwick, David Hill and Roger Smith, January 11-14, San Diego, California. (1998) 153–158.

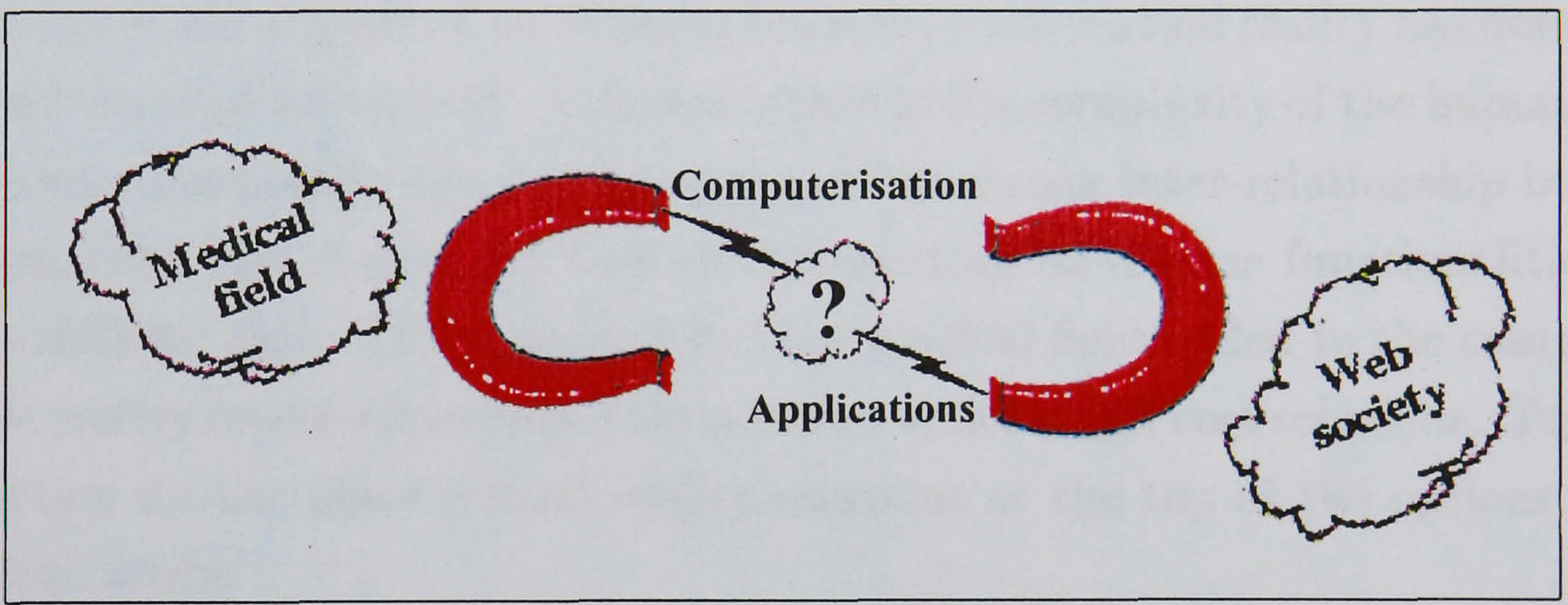
Contents

List of Figures

List of Tables

Chapter 1

Introduction



1.1 Research justification

The medical field is going through a computerisation process in order to accommodate the new information age. With this process a new discipline called *medical informatics* has emerged. This discipline deals with the storage and retrieval of biomedical information for the purpose of problem solving and decision making [157]. Under this general theme comes branches such as medical imaging, computer aided surgery, electronic medical records, .. etc. What all these fields share is the utilisation of computing and communication technologies to produce better health care. Training and education of medical staff is an area where computerisation is needed, especially in surgical training. This is due to the fact that surgical training is an expensive and time-consuming process. In practice trainees learn surgical procedures by watching consultants perform. Hands on practice- using animals or models rather than patients- is minimal and unrealistic. Indeed, the real practice occurs on patients. Consequently, newly trained surgeons are more susceptible to complications during a procedure, which forms an added risk. Furthermore, to evaluate the trainee, paper exams, subjective evaluation by consultants or video monitoring of procedures are used. Hence, the current situation in surgical training is as follows: there is spending of health care budget and consultant time for training without any systematic means of validating that the trainee has acquired the needed skills until they perform on patients!

The solution to this situation may come from medical informatics. The field of simulation jointly with virtual reality (VR) technology provide computer-based environments that can be used for surgical training. Research targeting this problem has been active since 1994. However, very few results from this research have been used clinically. One major reason is the sensitive nature of the medical field as a result of dealing with human subjects rather than objects. High fidelity is a requirement of any suggested substitute; however, so far virtual reality has not shown enough evidence of this aspect. Another reason is the complexity of the human body (its anatomy, physiology and psychology) and the strong inter-relationship between its components; this makes the task of representing its diverse functionalities and shapes a difficult task. The complexity of the medical field added to the complexity of virtual reality interactions push efforts in this area to high cost solutions. Together these factors do not place virtual reality solutions at the top of the options list in the clinical world.

At the same time, the World Wide Web (WWW) started to take a more formal shape, when the World Wide Web Consortium (W3C) was founded in 1994. Since then the WWW has become a virtual society that hosts a huge amount of information. The web society has been growing exponentially, as well as changing shape. It is redirecting its attention to provide services in addition to information. These services cover users with different needs, from highly specialized complex applications serving experts to applications of general utility such as e-commerce. In order for the web society to be useful to as large a sector of people as possible, various application fields must be provided. Naturally, the medical field would be one excellent target for the web society. Actually we can see the attraction between these two fields yielding some useful results already. Medical informatics has realised the potential of the web as a media, which has resulted in a number of medical web sites that target either patient-specific issues, general health interests or medical practitioners' interests. Nevertheless, the potential of the web in the medical field has not yet been fully exploited.

1.2 Research objectives

The need for computer-based training systems in the medical field has directed research in the computer industry to provide solutions. These solutions are mainly complex systems which require high-end workstations with multiple processors, dedicated graphics and special peripheral devices. As mentioned earlier, the reason for this high specification is the complexity of the application alongside the high computational requirement of VR solutions which promise realism. In this work we are providing a solution with a different flavour. Platform independence and ubiquity are two distinctive features of our proposed solution. We propose to deliver these two features by utilising the web environment. Because of the ubiquitous nature of the web, the solutions provided on its environment are expected to be accessible via low-technology equipment as well as high ones. Thus, there is a trade off between accessibility (provided by the web) and realism (provided by VR). Hence, comes the overall picture of this work- Figure (1.1).

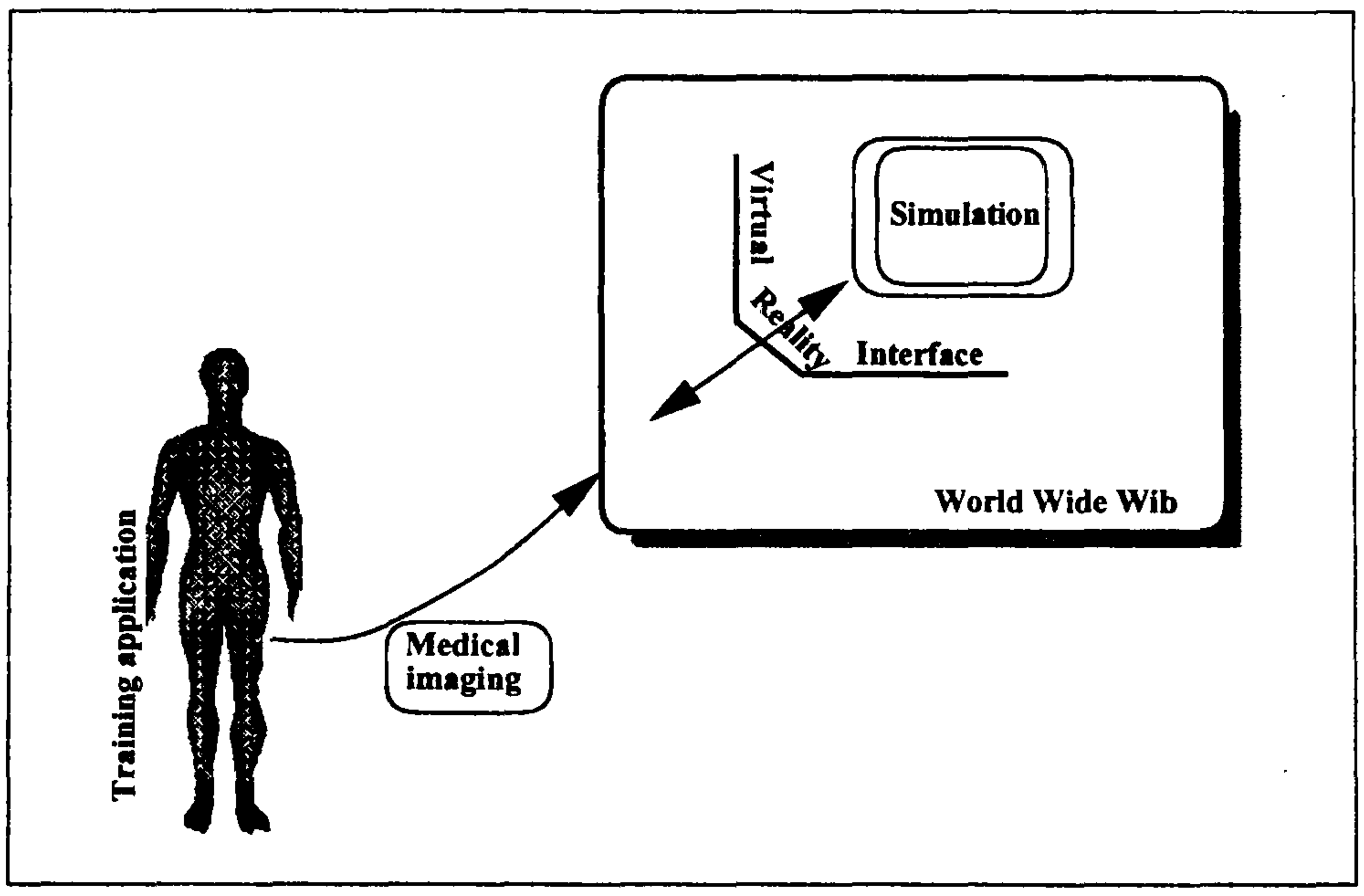


Figure 1.1: Overview of the disciplines involved in the work

In this work we investigate the feasibility of providing a surgical *training system* on the *World Wide Web*. In order to achieve the objective of the system- which is to transfer surgical skills- we need to provide a realistic interactive environment which reflects the real life procedure. *Simulation* techniques are used to build a model of this environment, while *virtual reality* technology is used to interact with the model. In the next section, we will define these basic concepts.

1.3 Basic concepts

- **Simulation**

  Simulation is the process of creating an approximate model of a system, or a process. The simulation model needs to be a flexible imitation of the process rather than a one-to-one copy of it. The purpose of simulation might be to understand the behaviour of this system or to investigate a hypothesis under this system.

  The simulation process has to be validated to assess how adequately it reflects the modelled process. Validation can either be done by comparing the result of the simulation with the real world one, alternatively, the result of simulation can be validated against the theory that was used to build the simulation [112].

- **Training system**

  The objective of a systematic training system is to ensure that certain skills, tasks or procedures are learned and satisfactorily transferred into the desired environment. There are three basic methods to transfer skills: the verbal method- telling the trainee what to do; the demonstration and guiding method- showing what to do; finally, the practical method- letting the trainee practice what to do in a constrained environment [87].

  It is essential to have some measure of transfer of training in the system, since it is easy to create a training system that transfers no skills, or a system that transfers negative skills. The positive transfer of skills in a training system depends on the contents of the training as well as the conditions of the training environment.

- **Virtual reality**

  Virtual reality (VR) is the technology that allows the representation of a real or abstract model such that it can be interactively experienced and manipulated by the participant. The objective of the virtual reality technology is to change the way humans interact with the computer into a more intuitive method, in anticipation that this will increase the human awareness of the modelled environment.

- **Web environment**

  The World Wide Web is a distributed information retrieval system that is built on top of the Internet- a set of interconnected networks which include commercial, university, research and military networks. The web environment has evolved both in size and functionality. Initially, the web environment was based on hypertext and the only functionality available was to retrieve information that had been set up on some server. Hence, one-way communication from the server to the client was the only connection possible. Later on, the web environment started to offer some simple forms of interaction when the Common Gateway Interface (CGI) was introduced. Currently, the information formats offered on the web are varied- e.g. audio and video streams and 3D environments. In addition, new themes and tools have emerged to implement distributed applications on the web.

1.4 Contribution of the proposed work

In order to provide a training system, analysis of the simulated process has to be done to identify the requirement of the simulation model. Furthermore, when virtual reality technology is utilised to provide interaction with the model, one expects another set of requirements which VR imposes to deliver the expected complex interaction. The situation gets even more complex when the web is used. It is still debatable whether the web will merely be another environment for applications or if it is going to shape the applications provided on it. However, there is more evidence that the web will demand its own set of requirements.

Our proposed surgical training solution joins these three disciplines- training, VR and the web- together; hence we have to deal with the three different sets of requirements. In this case, the following questions are investigated:

- How much do we have to compromise the different sets of requirements to accommodate others?
- How realistic, accurate and useful is the final system produced from merging these fields?

These questions are addressed in this work with focus on surgical training systems for one type of minimally invasive surgery- known as interventional radiology.

1.5 Thesis overview

Chapter 2 - Virtual reality in medicine - this chapter introduces virtual reality technologies and reviews the role of VR in the medical field. Some of the applications reviewed in this chapter target a similar medical area to our work, therefore offer different solutions to the same problem. Other applications focus on different medical areas, however they contributed to concepts and implementation that have been used in this work. Furthermore, this chapter includes VR requirements for surgical training systems.

Chapter 3 - Motivating application: Abdominal aortic aneurysm - describes the practical application that motivated this work and that is used to apply the combination of concepts- training, VR and the web. This chapter will analyse this surgical procedure and deduce our training requirements.

Chapter 4 - WebSTer environment - introduces web concepts and technologies that allow the building of web-based applications. In addition, the chapter includes a classification scheme and a review of the existing web-based applications. The requirements of the web alongside web-related aspects of our surgical training system are reported in this chapter. Finally, we conclude the chapter with an evaluation of how much did the web affect the VR and training requirements.

Chapter 5 - WebSTer user interface - describes interaction methods and technologies used in virtual environments. We set the requirements of our user interface based on the training requirements. A description of the user interface is then reported, followed by an evaluation of the provided solution.

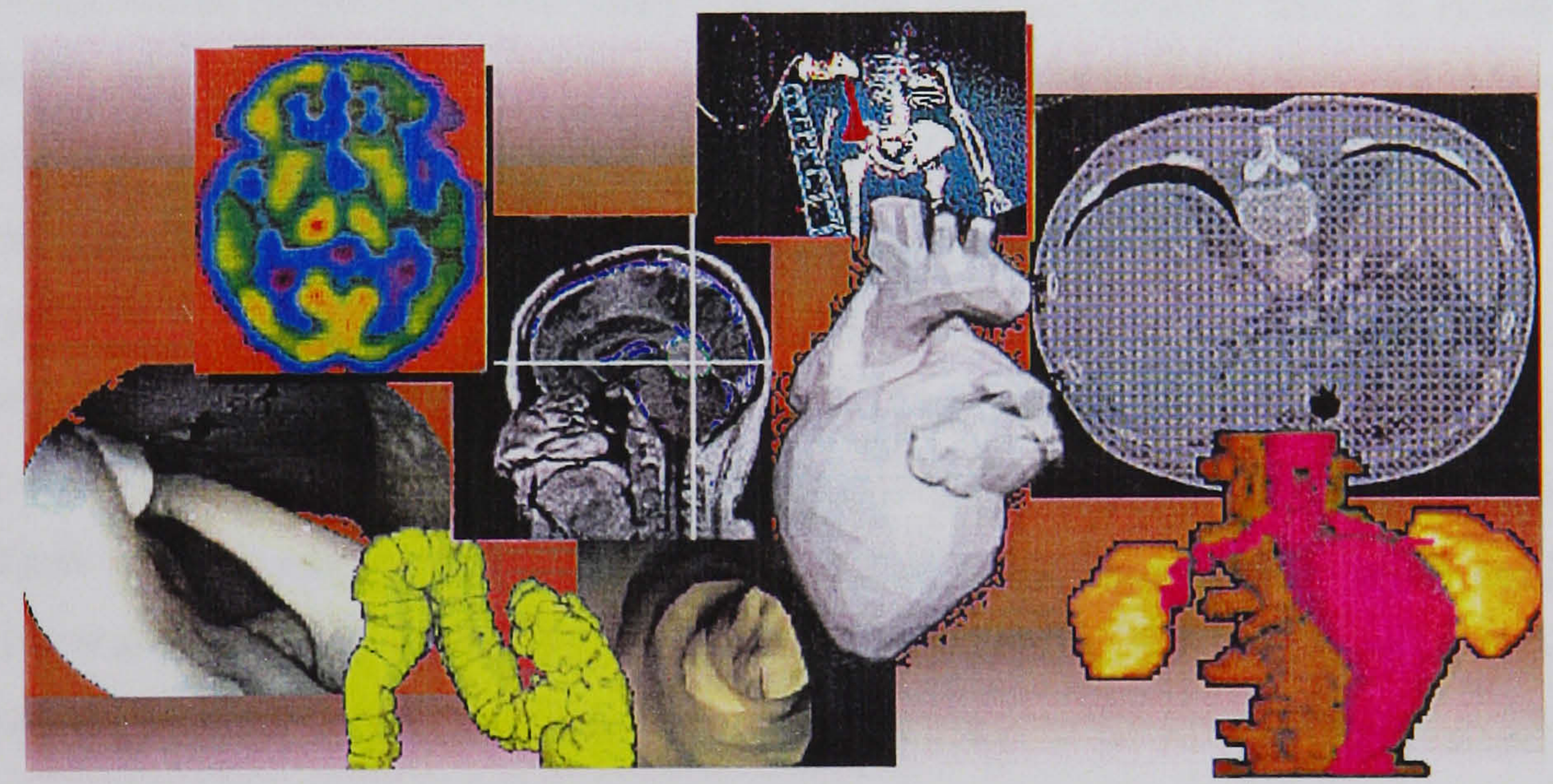
Chapter 6 - WebSTer modelling - this chapter is concerned with creating a geometric and behaviour model of the training environment. The theoretical background of how to create a model and the necessary mathematical concepts are reported in this chapter first. Then a description of the design and implementation of our model follows. We then conclude the chapter with an evaluation of our chosen model.

Chapter 7 - WebSTer evaluation - this chapter will discuss the overall system to evaluate our second hypothesis of how useful it is to merge the three disciplines: training, VR and the web. Radiologists, surgeons and medical visualization experts have been asked to evaluate general and specific aspects of the system. This chapter includes the questionnaire used and the results of the evaluation.

Chapter 8 - Conclusion and Future work - includes a summary of the reported work and suggestions for future improvements.

Chapter 2

Virtual reality in medicine



The unification of medicine with computer and communication technologies will transform medicine to "Information with medical flavour" according to Satava [151]. This union has resulted in the emergence of the medical informatics field. The purpose of this chapter is to introduce the reader to one aspect of medical informatics, namely the use of virtual reality and technologies supporting it. In section (2.1), we shall introduce virtual reality (VR) as a general concept. Then, in section (2.2), we will look at medical applications and how VR has been used to enhance these applications. Finally, we will identify benefits, requirements and limitations of the VR technology when applied to the medical field- section (2.3).

2.1 Virtual reality

Our interaction with the computer has always been an abstract metaphor of one sort or another. First, computers were thought to be big calculators. The keyboard was the means of entering text for data processing, and printed text was the output. Then, the graphical capabilities of computers were demonstrated in the early 1960's. At that time the idea of immersing the user with the computer output was introduced by Ivan Sutherland [100]. This idea was further researched secretly for military purposes. Meanwhile, the desktop became the widely accepted metaphor for human computer interaction and is still widely used today. The use of a monitor, keyboard and mouse is the way we interact with the computer. This interface represents a very simplified abstraction of interaction. The fact that it is abstract implies the need for initial training. It is true that the required training is little which is due to the other factor- simplicity. It is also due to this factor that our interaction with the computer is limited; it is limited to 2D devices attempting to present 3D worlds. Virtual reality (VR) aims to provide the solution. The virtual reality field is investigating new technologies to provide a more intuitive, natural and complex interface to the computer. By intuitive we mean requiring minimum training, while natural implies the use of day-to-day interaction methods such as gesture and talking.

Some people define VR as immersing the user in a multi-sensory computer generated experience[60]. This definition focuses on one aspect of VR while disregarding other aspects. If we are to examine the type of virtual reality systems available, we can see that they cover a spectrum which starts from interacting with reality and ends by interacting with synthetic virtual environments. Somewhere in the middle of this spectrum, the reality and the synthesis overlap to produce augmented reality- see Figure (2.1).

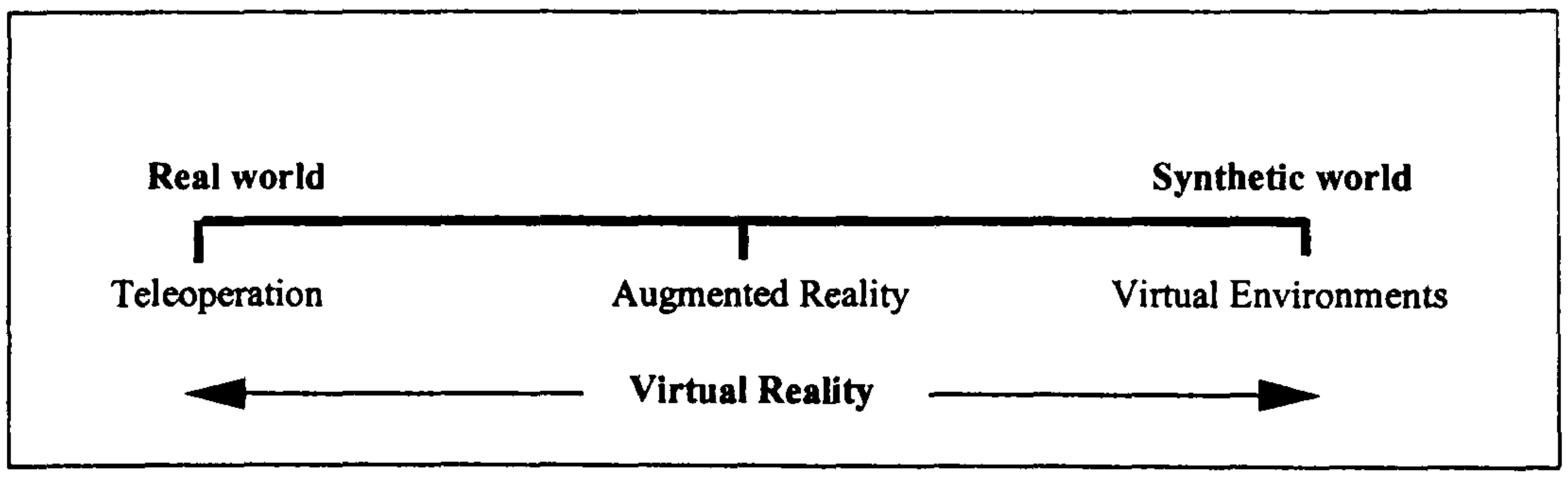


Figure 2.1: Virtual reality covers a spectrum from realistic to synthetic worlds

Augmented reality refers to a representation which mixes parts of the real world- either captured by video or directly perceived- with synthetic images. Virtual reality also provides a way of interacting with realistic worlds in what is known as *teleoperation.* A teleoperator is a "machine that operates on an environment and is controlled by a human at a distance"[31]. On the other hand, the user could be interacting with a fictitious world, totally generated by the computer, which we will refer to as a *virtual environment* (VE). As mentioned before, virtual reality provides a means of interacting with computer systems. This makes the VR field an extension of other existing fields [90]; figure(2.2) shows how simulation and robotics are extended to produce the three types of VR.

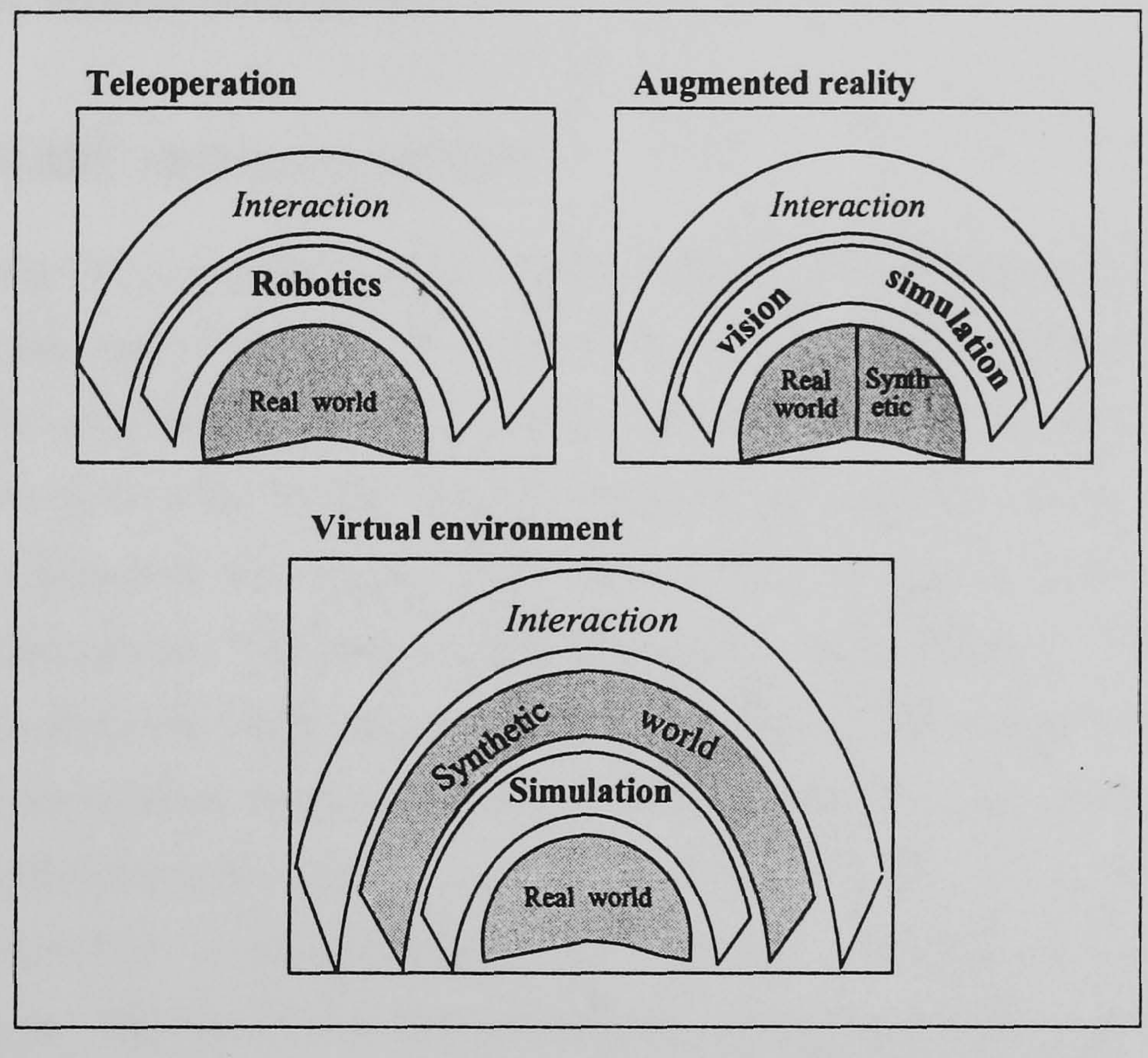


Figure 2.2: Virtual reality extends existing technologies

Virtual reality is thought to be useful in that it will minimise the learning effort needed to use computer applications, hence users from all disciplines will be able to easily make use of the computer capability in their work. Furthermore, it is thought to lift the burden of interacting with the computer from the users and allow them to concentrate instead on the information presented. Hence, comes the goal that VR is trying to achieve: making the user more aware of the presented environment [31]. There are two ways of achieving this goal. Some people think that engaging the user's senses with output from the environment will achieve the feeling of being in the environment. This philosophy is known as *sensory immersion.* On the other hand, *cognitive immersion* aims to achieve presence by engaging the user's thoughts- in the way that a movie does [80]. The supporters of the sensory immersion philosophy try to provide a three dimensional stereoscopic environment with audio and haptic feedback, which surrounds the user. They investigate the required hardware and software to achieve this environment. Meanwhile, cognitive immersion supporters look into how the human experience is built and try to understand how the perceived stimuli affect this experience. The fact is that both sides of immersion are needed and the degree to which each side is needed depends on the application. The type of VR application is not the only influencing factor on the technology used. Stanney suggested that the characteristic of the user- e.g. abilities and experience- as well as the characteristics of the human perception system are two factors which can affect the potential of virtual reality[165].

2.1.1 Virtual environments

In this work, we are concerned with synthetic virtual environments- which we will refer to as virtual environments (VE). In order to build virtual environments, four components are involved which reflect four different areas of technology involved in VE. The first component is modelling the virtual environment; this activity involves geometric description of the virtual objects in the environment, their behaviour and their inter-relationships. The second component is visual display. This involves providing hardware and software to accommodate how the binocular three dimensional human visual perception works. Visually coupled systems- e.g. head mounted display (HMD) and binocular omni-oriented monitor (BOOM)- are examples of such technology, where the visual perception of the user is tracked to achieve a natural stereoscopic view. Similarly the third component, audio display, accommodates for the characteristics of sound which affects its perception and cognition. The head related transfer function (HRTF)-which represents the change of sound as the source moves- is used to facilitate the localisation of the source of sound. The last component deals with haptic technologies which allow us to perceive objects via our touch sense. The haptic sensation is displacement of the skin which could be extended, transient or frequently transient. This sensation is composed of two elementary feelings: tactile and kinesthetic. Tactile refers to the spatial pattern of pressure on the skin which allows the identification of the surface texture. Meanwhile kinesthetic refers to the relative position and movement of the body parts as well as the muscular effort [31, 60]. Figure (2.3) shows the components of the virtual environment.

In immersive VE, these components are integrated to give a feeling of presence in the environment. Some senses are redundantly used to support another essential sensation. For instance, one might provide a door squeaking sound when the user opens a door to enhance the feeling of presence in the environment[165]. In other

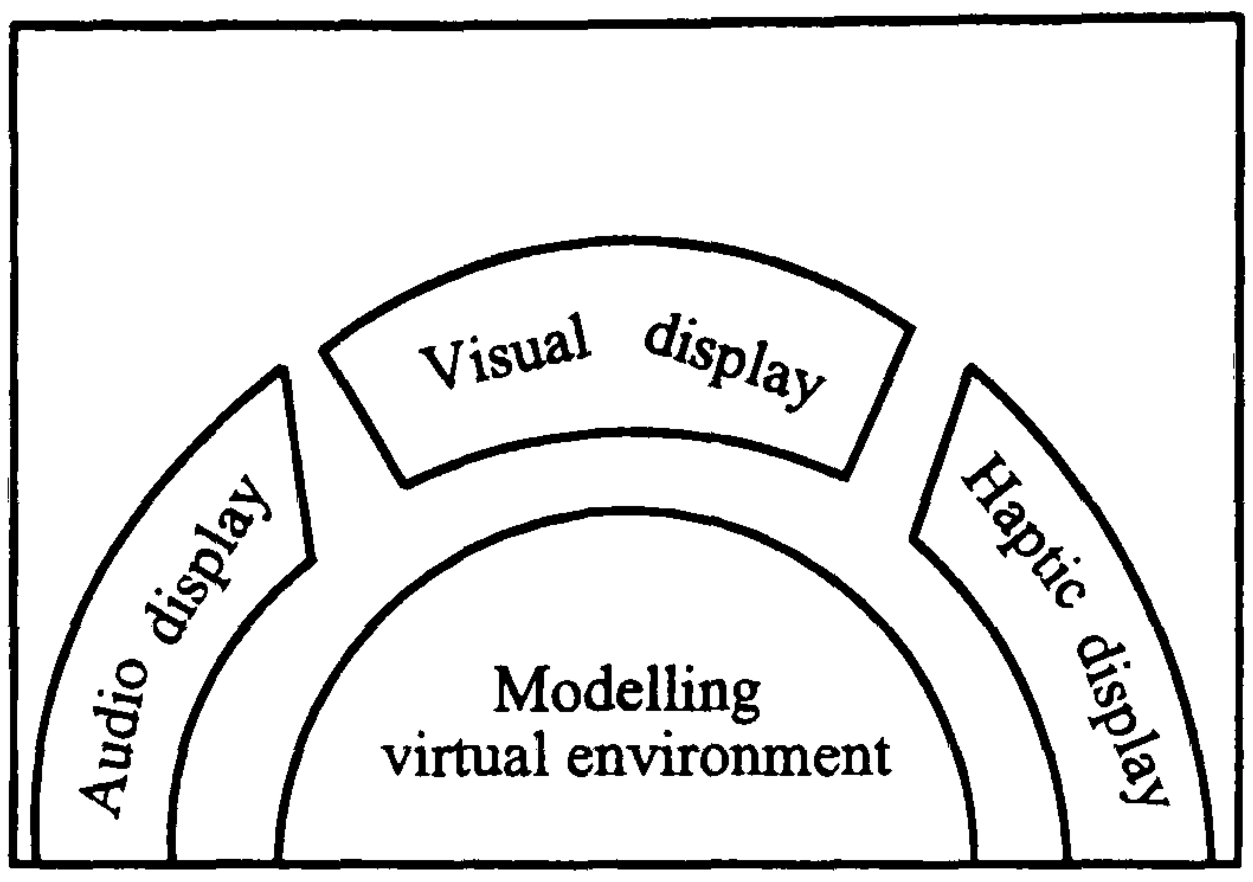


Figure 2.3: components of the virtual environment

situations, the application requires special type of sensation that is essential for the successful simulation of the environment. For instance, in minimally invasive surgery, the haptic sensation is an essential component of the procedure. Hence, the degree by which a virtual environment component is involved in the creation of the environment depends on the requirements of the application.

Virtual reality is characterised by rich interactions. To facilitate this feature multiple input and output channels are used as well as high degrees of freedom (DOF) of these channels. Hence, VR has been responsible for pushing for the development of a number of new peripheral devices. For instance, display devices in VR have focused on providing 3D stereoscopic and sometimes immersive images. The utilisation of shutter glasses alongside with time-multiplexed or split windows monitors provide the stereoscopic effect with a normal display, hence constitute a low cost option. HMD and BOOM display units provide immersive stereoscopic views, however they come with a list of disadvantages- e.g. discomfort and sickness. The Cave Automatic Virtual Environment (CAVE) is another option for immersive display, where the user is placed within a room with projection screens on the walls and sometimes the floor and the ceiling. This option avoids the problems of the HMD, however it is expensive. There are other options which are suitable for multiple user display, such as the wide screen and the workbench with shutter glasses. The workbench is a projection screen which is horizontally reflected onto a table; this metaphor is suitable for some fields such as surgery [67, 31].

On the other hand, input devices have evolved to cover higher DOF while some allow direct manipulation of objects. Spaceball, joysticks and 3D mouse are all examples of 3D devices which are currently affordable. Data gloves detect position, orientation and finger flexion angles. Head and eye tracking are also used to close the visual interaction loop, giving the user the sense of being surrounded by the environment. Devices can be tracked either using mechanical, magnetic, acoustic or optical methods. Acoustic and mechanical devices are not very accurate, while magnetic methods suffer from limited working range and are prone to distortion. Infra red LED- light emitting diodes- have been used to track head movement, but they also suffer from limited working range [100, 26].

Haptic feedback devices have been developed to provide both force feedback and tactile feedback. The latter can be done using small electrodes which are hooked on the data glove or using micro pins that vibrate based on an electric current. Meanwhile, force feedback devices are more popular and vary according to the number of fingers they affect and the DOF detected. These devices can be body-based, such as exoskeleton devices, or desktop devices, such as the Impulse Engine- a joystick like device- and the PHANTOM- a pen like device affecting one finger [193].

VR technology has been used in several fields, such as engineering design, military applications and entertainment. Furthermore, virtual environments are thought to be an excellent environment for training and education. They provide a safe, systematic and rich environment to transfer skills and knowledge. It is this training aspect of virtual environments that we are interested to utilise, hence we will take a closer look at it in the next section.

2.1.2 Virtual reality in training

Virtual environments which mimic real situations can be built and used to train subjects on certain tasks. In industry, this could be used to train workers on handling machines, installation and fixing faults,..etc. In medicine, physicians can be trained on surgery/or nurses can be trained on certain procedures or tests. Theoretically, all sorts of skills can be transferred through using this concept. However, in practice the current VR technology is far from this ideal position. Experimentation has been done to verify the capabilities of VR in training. Kozak [107] and Kenyon [103] carried out separate experimentation to test the transfer of motor skills through a virtual environment. The experiment involved a number of coloured disks arranged in two rows and five cans placed on the front row of the coloured disks. The task involved moving the cans from the first row to the second row and placing each on the disk with corresponding colour as the original disk. In Kozak's experiment the user was immersed in the virtual environment using a head mounted display, while data gloves were used to track the hand motion. Meanwhile, Kenyon used a CAVE and gloves for interacting with the virtual environment, where the trainee was able to see his hands and part of the room as well as the synthetic environment. Kozak reported that subjects trained using the virtual environment did not show any transfer of skills to the real world. On the other hand, Kenyon reported that virtual environment trainees showed some improvement in performance in the real world. Although Kenyon's results were more in favour of virtual environment training, they both recognised a number of factors that hindered the transfer of skills through the virtual environment. First, the lack of tactile information in the virtual environment made the trainee totally dependent on their visual perception to accomplish the task, which made the time required to complete the task in the VE longer than the real world. Second, the trainee had to put up with lag in the VE which made the virtual hand inconsistent with the actual motion. Such inconsistency turns the training environment from an environment to transfer skills to the real world to an environment to train on virtual environments! One reason that Kenyon's results were better than Kozak's is the ability of the trainees to see their hands, which gave them better coordination between the virtual environment and their movement. Furthermore, they both recognised that the enrichment of the environment with realistic appearances, depth cues and sensory stimuli would improve the transfer of skills. In addition, Kenyon tested for the transfer of strategy skills in his experiment; however, he found little evidence of its transfer. It is thought that such skill would require more time to transfer than the tested time.

In another experiment, Regian [139] tested for the transfer of procedural and representational skill using a virtual environment. The procedural skill was tested via a missile launching sequence, while the representational skill was tested using the navigation task in a building. Regian compared the performance of subjects trained using a virtual environment and using a computer-assisted instruction system (CAI). The purpose of these experiments was to verify the transfer of skills over a virtual environment, as well as to test if it would provide an extra degree of transfer compared to the other training methods. Regian used a head mounted display and data gloves for the immersive virtual environment. The result showed that there is a positive transfer of skills in both environments. However, no significant difference was noticed between the virtual environment trainee and the CAI trainee in performance.

The previous experiments seem to suggest that VE are not significantly better than other training methods in transferring skills to real life. If one adds to this the current high cost of building and providing VE, one would most likely choose other training techniques. However, these experiments should not be used to judge the suitability of virtual environments for training. This is due to the fact that VE technologies are not yet mature. In fact, in each of the reported experiments, authors report faults or limitations in the virtual environment which they are aware of and know that it limits the virtual experience. What these experiments show is that the virtual environment used in training should take into account the type of skills involved in the task and the best methods to transfer these skills. For instance, Kenyon noticed that augmented reality helps in training motor skills because the trainee is aware of his own body movement as well as the environment. Another example is Regian's observation that bird's-eye perspective has helped in the transfer of the representation skill in the navigation task more than the immersed perspective. It is this type of result that should be aggregated from such experiments to improve our understanding of how skills are transferred and consequently enhance the design of our virtual environments.

2.2 Medical applications: Review

The medical informatics field consists of three components:

- Data
- Methodology
- Theory and knowledge

The first component- Data- is concerned with acquiring patients' medical data, such as patient records, laboratory results and medical images. A huge project is currently ongoing under the supervision of the Health Informatics Planning Panel (HISPP) in USA to develop standards to specify the clinical data of the patient and the medical knowledge to build Electronic Medical Record Systems (EMRS)[16]. In addition, this component includes the medical imaging area where digitized data sets which show different types of information about the human body are collected. Section (2.2.1) will look in more detail at this area.

Medical informatics is interested in computerising the different methodologies of the diagnosis and treatment process. The aim of this is to utilise computer technology to assist, teach or perform these methodologies. Recently, a number of applications have been developed which prove that computer technology is able to fulfill these three goals. Virtual reality technology has been extensively used in these applications. Indeed, medical informatics can benefit from all types of VR.

For instance, both virtual environments (VE) and augmented reality (AR) are used to support diagnosis and treatment. Likewise, teleoperation can be used to diagnose and treat remote patients in rural areas and battle fields. In this section, we will review some of these applications in such a way to cover the three branches of the medical process: diagnosis, treatment and training. The applications reported here serve a dual purpose: one is to give examples of efforts in this field and the second is to compound knowledge that will be used later in our own solution- section (2.3) will present a list of these lessons. In section (2.2.2) we will report applications which use VR to support diagnosis. In case of surgical treatment, these technologies can assist in planning the procedure- as will be shown in section (2.2.3), as well as assist in performing it- section(2.2.4). Applications which utilise VR for medical education and surgical training will be presented in section (2.2.5) and (2.2.6) respectively.

Finally, in section (2.2.7) we will briefly discuss the theory and knowledge involved in the medical informatics field. Figure (2.4) shows a road map of this section, with the section in which different topics are discussed.

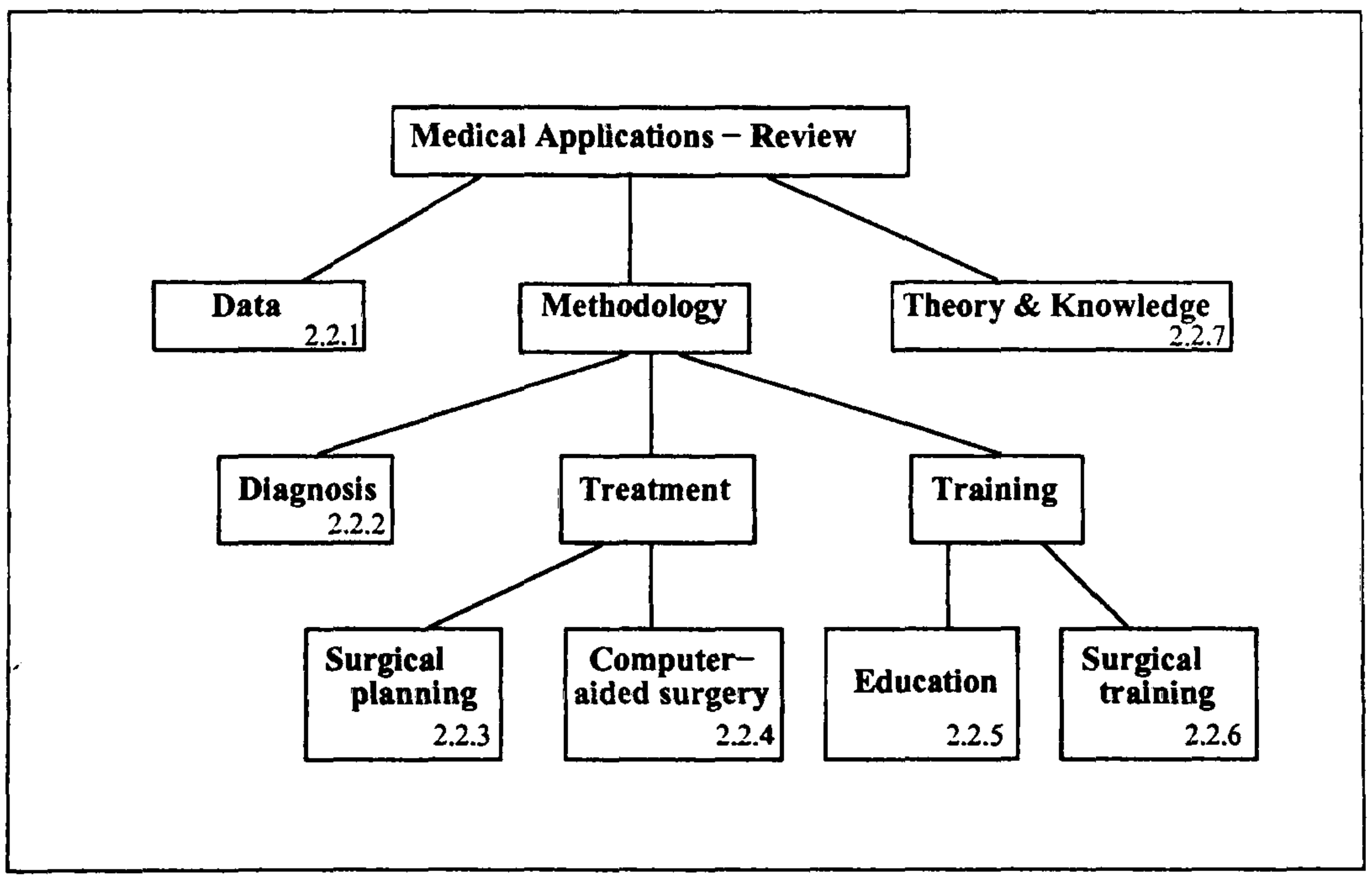


Figure 2.4: Medical applications - road map

2.2.1 Data acquisition: Medical imaging

Nowadays physicians rely heavily on medical images from multiple modalities to diagnose and treat patients. Recently, medical imaging has advanced to provide several techniques which measure different physiological and anatomical properties of the human body. X-rays have been used for a long time to identify bone structures by integrating the attenuation of the radiological signal along the ray to produce a static image. Another type of X-ray imaging technique- called fluoroscopy- captures sequences of images to produce a movie of the anatomy. Fluoroscopy is used nowadays to guide some surgical procedures, where the X-rays are recorded after they pass through the body using an "image intensifier", and then shown on a TV screen. Because of the digitization of the X-ray measurements, it is now possible to enhance and manipulate X-ray images. For example, these images can be subtracted from images taken at different times to show the flow of a contrasting agent such as iodine. This type of imaging- known as digitized subtraction angiography (DSA)- is used to enhance the visualization of blood vessels in the body [20].

A series of X-ray cross sections along the body a few millimetres apart can be acquired and reconstructed to give another imaging modality known as computed tomography (CT). In spiral CT scans, the beam source and the detector are rotated around the longitudinal axis of the patient, while the patient slowly moves through the X-ray equipment providing continuous acquisition of the anatomy. A CT scan provides images which show bone and tissue structures of the body; however different tissue types are not clear in CT.

Magnetic resonance imaging (MRI) is another modality which is able to measure different properties of soft tissues in the body- e.g. proton density. MRI measures the radio frequency energy of tissue cells when exposed to perturbation of the magnetic field. Magnetic resonance angiography (MRA) is an MRI technique that is able to show blood vessels non-invasively [140].

Ultrasound is another imaging modality which depends on measuring the acoustic impedance and delay of ultrasonic waves when reflected from tissues. It can produce real time images of the body, showing soft tissues that are not surrounded by bone. An extension of this technology- called colour doppler sonography- analyses the ultrasound waves to show the blood flow in the vessels coloured and superimposed on the normal ultrasound images.

In nuclear medicine, radioactive isotopes are injected into the body. The distribution of these isotopes measures different functionalities of the body such as blood flow and neurological stimuli. Measuring the metabolic activity of the organs helps to detect cancer cells (high metabolism), hence nuclear medicine can be used to detect and monitor the treatment of cancer. Imaging modalities such as SPECT- single photon emission computed tomography- and PET- positron emission tomography- are examples of nuclear imaging modalities [127]. Once these digital representations are acquired, they form the basis for diagnosis and treatment methodologies.

2.2.2 Diagnosis

Virtual endoscopy is an example of a computer based technique, which can be useful in diagnosis. The idea of virtual endoscopy is to use non-invasive imaging techniques such as CT and MRI to replace invasive diagnostic procedures such as endoscopy. In endoscopy procedures a long tube with a camera mounted on the tip is inserted into hollow parts of the body such as blood vessels, colon and bronchial tree. The images captured by the camera are used to detect polyps on the surfaces. The advantage of virtual endoscopy is that it saves the patient from the discomfort of the invasive techniques. In virtual endoscopy, 3D models from CT or MR data are constructed using either surface techniques- as in [89, 126]- or volume rendering- as in [138, 148, 152]. The physician then is given the facility to fly through the data to explore it and detect abnormalities. Systems that implement virtual endoscopy provide various methods of exploring the data. In [148, 126] mouse and 3D input devices are used to navigate through the structure, while automatic path planning is offered in [116, 138, 152]. The virtual endoscopy system provided by State University New York at Stony Brook offers more sophisticated navigation options. A potential field is used to prevent collision between the camera and the surface of the organ, while attraction forces are used to make the camera motion preferably at the centre of the lumen and heading towards a pre-specified target. In addition, the user can refine the path by mouse clicking on the camera to explore other parts in more detail. This mechanism allows the user to navigate through the structure without the danger of getting lost in the outside space as well as being able to zoom into areas of interest without losing the overall path [89]. Furthermore, they provide several views of the anatomy explored: an endoscopic view, an overall view and three slice views- sagittal, coronal and transverse. So far, virtual endoscopy has not been used clinically, since some information required for the diagnosis process- such as tissue texture- is not acquired. However, volume rendering techniques are preferred to surfaces since the amount of information lost is minimal.

Augmented reality has been also utilised to assist the physician in diagnosis. For instance, the North Carolina augmented reality system projects images of the womb from ultrasound on the patient. Physicians are able to localise the ultrasound within the natural space which assists them in understanding the images[25].

On a different scale, teleradiology is an area where computer and communication technologies can help in diagnosis by joining human expertise. The idea of teleradiology is to share the patients' medical images between experts separated by any distance to exchange opinions and expertise. Silicon Graphics proposed the use of shared workspaces to allow the physician to share 3D models of patient data[43]. A similar idea has been proposed in [50], where a volume visualization package-called InViVo- has been extended to share images over a TCP/IP connection. In these applications the participants share annotator and white board, and use voice to collaborate.

2.2.3 Surgical planning

Specialised systems have been developed to plan surgical procedures. For instance, a system is presented in [131] which helps the physician plan a hip replacement procedure. Computer graphics have been used to build an environment where the proper hip size and the required fit-force is the outcome of the rehearsal. The finite element method has been used to predict the stress and strain resulting from forces. These results require one day of computation to obtain them.

Virtual environments have been developed for similar purposes. In [106] a system that helps surgeons in replacing fracture of the mandible is presented. Using stereoscopic shutter glasses and data gloves the physician can manipulate the bone fracture. The system evaluates the success of the plan according to smoothness of the skin. Another system presented in [37] utilises virtual reality technology to investigate alternative operations for hip and knee orthopaedic surgery. The system provides a fusion between CT and NMR (nuclear magnetic resonance) data to explore the case. The user can perform measurements in the traditional two-dimensional way or in three dimensions using a spaceball.

2.2.4 Computer - aided surgery

Augmented reality has been utilised to provide intra-operative assistance to the surgeon. Most intra-operative assistance takes the form of fusion of intra-operative data of some form with pre-operative images or planned path to facilitate the accomplishment of the target. For example, in the virtual reality assisted surgery programe (VRASP) carried out by the Biomechanical Imaging Resource in Mayo Foundation, the surgeon is provided with pre-operative data superimposed on the patient using a HMD. Customised surgical instruments and data gloves are used to track the hands of the surgeon [141, 142]. Similarly, an image-guided surgery system presented in [56] utilises HMD and magnetically sensored surgical tools to correlate between intra-operative MR scans and pre-operative CT scans as well as showing the position of the tools.

Alternatively, intra-operative assistance has been provided using desktop virtual reality techniques. For instance, in a project carried out at the University of Tokyo, the surgeon is provided with a planning system to find the minimum invasive path- i.e. minimum damage to healthy tissues- for needle insertion in a neurosurgical procedure. The surgeon is offered assistance during the actual procedure to help localise the needle relative to brain structures. The system provides a stereoscopic view of the brain (taken from pre-operative CT or MR data) with the current location of needle which is tracked using a 6 degrees of freedom (DOF) mechanical manipulator [118]. In addition, in [49] the VISLAN system provides the surgeon with pre-operative data and video captured during the operation. The pre-operative data can be shown imposed on the video monitor while a computer workstation shows three views of the pre-operative data with the position of the tool indicated on each slice as well as on a 3D construction of the pre-operative data. Finally, the ARTMA virtual patient system provides intra-operative assistance to the surgeon during endonasal sinus procedure. The intra-operative video from the endoscope is blended with pre-operative images. The position of the catheter is tracked using magnetic fields. The pre-planned path is shown super-imposed on the video monitor; when the surgical tool drifts from the planned path the image of the path is distorted to indicate the drifting [79].

Because of the critical nature of intra-operative applications, all these systems must be extremely accurate in the data they present and in the registration between intra- and pre-operative data. It is essential that the interface with these systems does not add extra burden on the surgeon since the goal is to assist them. For instance, operation time should be maintained constant if not reduced by the utilisation of these systems. When these two requirements are achieved, such systems add an extra safety factor to the procedure.

It is unlikely that computers will replace human medical operators in surgery. However, human operators are not highly accurate when it comes to positioning or tracking a trajectory. It is in these areas that computers are used to replace human operators. Robot manipulators are now used in procedures that require high accuracy such as neurosurgery. Robots are used to position an instrument to millimeters accuracy and to advance these tools according to a specific trajectory [104]. Similarly, in microsurgery-e.g. eye surgery- robot arms are used to apply the surgeon hand motion after filtering the tremor of the hand and sometimes scaling down the forces [93].

2.2.5 Education

Computer and communication technologies can introduce new efficient methods of medical education. Medical education covers a range from teaching students anatomy to training surgeons on specialised procedures. We will deal with the latter in the next section.

Recently, virtual reality technology, multimedia capabilities and computer simulators have been utilised to present new methods in medical education. For example, an ongoing project being carried out in the University of California, San Diego is building a networked educational system which utilises the current computer technology from multimedia to virtual environments. The project will link local knowledge from textbooks, anatomical atlases with the web environment. It will incorporate electronic medical records and online simulators with different levels [86].

On the other hand, Kancherla [102] utilises augmented reality to teach students how bones move when the body moves. The 3D model of the skeleton is seen imposed over a video capture of the patient while moving. A generic 3D model is used, however scaling is used to make it fit to different patient cases. This model is driven based on kinematics.

2.2.6 Surgical training

A number of medical simulators have been developed to train surgeons on new and existing surgical procedures. Two different approaches have been taken to provide these solutions. One approach is to develop a general toolkit that can be used to build different medical simulators. On the other hand, others provide application-specific simulators that can be used to train on certain procedures.

Under the first category is a package developed by Kernforschungszentrum Karlsruhe (KfK) called "Kinematic Simulation, Monitoring and off-Line Programming Environment for Telerobotics" (KISMET). The package allows interaction with the models using shutter glasses and sensored surgical instruments. The package can model the deformation of tissues. It will also allow the remote manipulation of instruments over a LAN- local area network. This package has been used to develop a training system for minimally invasive surgeries [83, 108].

Ixion is another general toolkit that provides the facility to build 3D anatomical models and add deformation and collision detection. The package can provide various patient-specific data with different pathology. Surgical instruments are sensored and force feedback is provided as output [88].

HT Medical has developed an authoring tool kit called Teleos to assist in modelling surgical simulators. The tool kit and the simulators run on a multiple processor SGI Reality Engine machine with dedicated graphical hardware. The package allows the representation of data as deformation spline-based models, volume, polygonal mesh or physically-based tubes. Blood flow can be modelled and controlled to simulate pathology. Deformation of organs is simulated and collision detection and response are provided. This package has been used to simulate laparoscopy, needle insertion and catheterisation procedures. In the catheterisation procedure the surgical tools- sheath, catheters and guide wires- have been modelled. The positions of the three instruments are tracked using custom interfaces. The simulator models the inflation of the catheter balloon and contrast agent injection which are typical in this type of procedure. Complications such as blood clots, balloon rupture and puncture of blood vessel are simulated as well. The simulator provides different options to treat the patient either by using clot-busting drugs or open clogs with the balloon or stent deployment [121, 120, 85].

On the other hand, a number of simulators have been developed to train physicians on a specific procedure- e.g. arthroscopy [113, 194], endovascular [180, 2], endoscopy [70], eye surgery [93, 135], sinus surgery [191] and minimally invasive surgery [11]. Five requirements have been thought to be necessary to build a surgical simulator [58, 150, 151]:

- Model the anatomy of the organs; it is essential that the model is visually accurate and realistic.
- Provide means of interaction with the objects in the environment.
- Input/ output sensory stimuli, where the number of sensor channels provided exponentially increase the sense of realism.
- Model physical properties of the organs.
- Model physiological properties of the organs.

Another requirement should be added to the previous list in case of a training simulator, which is providing some form of evaluation for the trainee's performance. The present surgical simulators vary in the degree they provide these requirements and the methods they use. In the next few paragraphs, we will review available training simulators based on these requirements.

When providing visual representation of the anatomy some simulators have chosen the volume representation [72, 191], while others use the surface representation. The advantage of volume data is that it facilitates tissue cutting and collision detection as well as preserving the information. On the other hand, surfaces are faster to render and they facilitate collision response.

In some simulators different surgical tools have been modelled [53, 135]. The degree of complexity of the model varied from one application to the other. In [23]- a catheter simulator provided by Nagoya University- the forward and backwards motion of the catheter has been modelled. Meanwhile, the bending behaviour is limited to constant angles and twisting has not been modelled. On the other hand, in the simulator provided by CieMed- called daVinci- which is used to train radiologists on endovascular procedures, the catheter has been modelled using the finite element method with beam elements to precisely simulate the catheter behaviour [2].

To facilitate interaction with the environment simulators provide input devices ranging from joystick [23], data gloves [130, 109], sensored surgical tools [135, 194, 113, 134, 2] and robot mechanical limb [93]. In [11] a frame holding two laparoscopic instruments with sensors attached is used to detect the position, rotation and functionality of the physical surgical tool, to reflect it in the virtual environment. In addition, some simulators provide force feedback- e.g. [93, 134, 135, 23, 72, 191]. In [23] experiments have shown that force feedback in addition to the visual feedback has reduced the force of collision of the catheter with vessel walls, which indicates that the availability of both sensations enhances the user performance. Meanwhile, [109] used sound as a replacement sensory stimuli for force feedback.

In some simulators immersion in the environment is obtained by adding a three dimensional tracked headset. For example, in [109] a head mounted display is used, while Oyama [132] used a BOOM. In addition, Delp [53] presented a desktop immersive virtual environment by utilising shutter glasses. In the daVinci simulator, a virtual workbench and stereo glasses have been used to provide three-dimensional stereoscopic vision [2].

Physical properties of the organs have been added in some simulators to model the behaviour of the anatomical structures when manipulated by the tools or by the user. In [191] deformation of tissues has been modelled by deflecting the rays which penetrate the volume data- in a ray casting algorithm- with deflectors with certain shapes and sizes. The accumulation of a number of deflectors can be used to model deformation and cutting in tissues. The finite element method has been used in [93] to model the elasticity of the coronal eye membrane. Similarly in [113] the finite element was used to model deformation of membranes in the knee. Finally, Cotin used the same method to model the behaviour of the liver [51].

The simulation of the pathology of the body is more problematic because it requires additional computer power. In [180] an abdominal aortic aneurysm training simulator models the effect of breathing on fluoroscopic images. The model is simplified to use 2D warping of the images to achieve this effect. They also model the injection of a contrast agent in the blood vessel using 2D image processing techniques- filtering and adding images. Meanwhile, in [53] a simulator has been presented to train users on traumatic injuries during military or civilian situations. In order to achieve the goal of this simulator, blood visualization as well as muscle tension due to injury are modelled in this simulator.

Finally, only three of the reviewed simulators have provided evaluation to their trainee. This is due to the fact that most of the current simulators are experimental and prototypes that are not ready to be used clinically yet. In [132, 133] the evaluation of the trainee has been achieved by capturing a video of the actual performance of the procedure. The physicians can then review and evaluate the performance of the trainee. In [130] a more automatic solution has been presented. In this simulator the trainee is asked to practice two of the tasks common in laparoscopy: suturing and dissection of blood vessels. A set of criteria for evaluation has been set- e.g. length of dissection, tightness of the knot and position of the suture, and then, fuzzy logic has been used to evaluate the trainee's performance. Meanwhile, the MIST/VR training system developed by Virtual Presence has been developed for the specific target of assessing the physical skills of the trainees to perform minimally invasive surgery. The system adopts simple graphics which does not attempt to simulate the anatomy of the body, but rather to convey the training skills. Six tasks are included with different complexity which require one- or two- hand grabbing and moving of objects. The physical interface- laparoscopy surgical instruments with sensors- of the system is tracked and these measurements are stored in a database. The measurements are used to assess the trainee's dexterity skills which are required for this type of surgery.

Some simulators provide other complementary features which can add more value to the training environment. For example, in the catheter simulator provided by Nagoya University, a teleoperation facility has been provided. Trainees can share the training system and interact over an ATM network [23, 24].

2.2.7 Theory and knowledge

In the last few sections we have reviewed a number of medical applications that have been developed for different purposes. We have noticed that they all share a high computation cost. This feature of medical applications has been the inhibiting factor for advancements in this field for several years. However, lately development in computer hardware, computer graphics software and algorithms have permitted the computerisation of the medical field- a good review of the advancements in computer graphics and its impact on medical applications can be found in [161]. Nevertheless, we still need significant computing power to meet the demands of this complex field.

However, computer technology is not the only thing that the medical informatics field require; knowledge is also lacking. In this field, knowledge represents the combination of medical knowledge such as anatomy, physiology, psychology,..etc. and human-computer interface theory. On one side, one would need to acquire the medical knowledge to make it available for computer technology. For example, anatomical structures of the body needs to be transferred to computer models. Similarly, the typical cycle of symptoms → diagnosis → treatment needs to be transferred somehow from the physician's brain to the computer. One attempt to achieve this goal is a project at Tokai University to build a "hyper hospital". The project is investigating the use of VR to automate the hospital visit experience such that it can be experienced remotely. The project will look at providing medical consultation via an intelligent system, examination and simple treatment [192].

On the other hand, one would want to mix the two sets of knowledge to better understand how we can utilise computer technology to provide better medicine. Computer graphics capabilities have been used to represent medical images in such a way that allows better understanding. Techniques such as volume rendering or surface rendering allow the representation of this data as 3D objects. Thus, the data sets can be manipulated in 3D, which is useful to understand the relationship between organs, spatially localise structures, and to detect pathological structures. There are a number of computer packages which provide various techniques to explore medical images such as ANALYZE from Mayo Clinic [143] and Voxel-Man from University - Hospital Eppendorf [99]. In addition, virtual environments have been used in [166] to explore medical data from different modalities. Stereo viewing, speach recognition and date gloves have been provided to assist students and physicians to gain better understanding of the anatomy.

Furthermore, virtual environments have been developed to assist physicians in understanding pathology. For example, in [82] a system has been presented to visualize the blood flow in an aneurysm. The system uses 3D models of the blood vessel constructed from DSA images. Partial differential equations have been used to model the behaviour of the blood- speed and pressure- on the aneurysm wall. The result of the behaviour model is then stored in a database. Different visualization techniques have been used to show these results including animation with the heart cycle.

Currently, our medical/computer knowledge is limited partly due to the complexity of the medical field and to our limited understanding of human-computer issues. This knowledge can be increased by extensive experimentation of computer applications in medicine. In fact all the applications previously reported in this section and others in the medical informatics field feed back to this knowledge. That is why it is crucial to evaluate the effectiveness of these applications from the clinical point of view.

2.3 Discussion

We can conclude from the virtual reality section- (2.1)- three general requirements of a virtual reality system:

- Realistic and accurate model of the simulated environment.
- Closed loop interaction.
- Special devices to achieve the required level of interaction.

The list of requirements presented in section (2.2.6) is an application of these general VR requirements to the medical applications. Furthermore, the previous medical review highlights some important features of medical applications, which will influence our solution:

- Three-dimensional representation and manipulation of structures help in gaining better understanding.
- Although 3D representation is advantageous, physicians are not used to doing things in 3D space. Hence, one should provide some form of linking between traditional and 3D methods- as we saw in the virtual endoscopy system presented by State University of New York- page (17)- and the surgical planning system for orthopaedics- page (18).

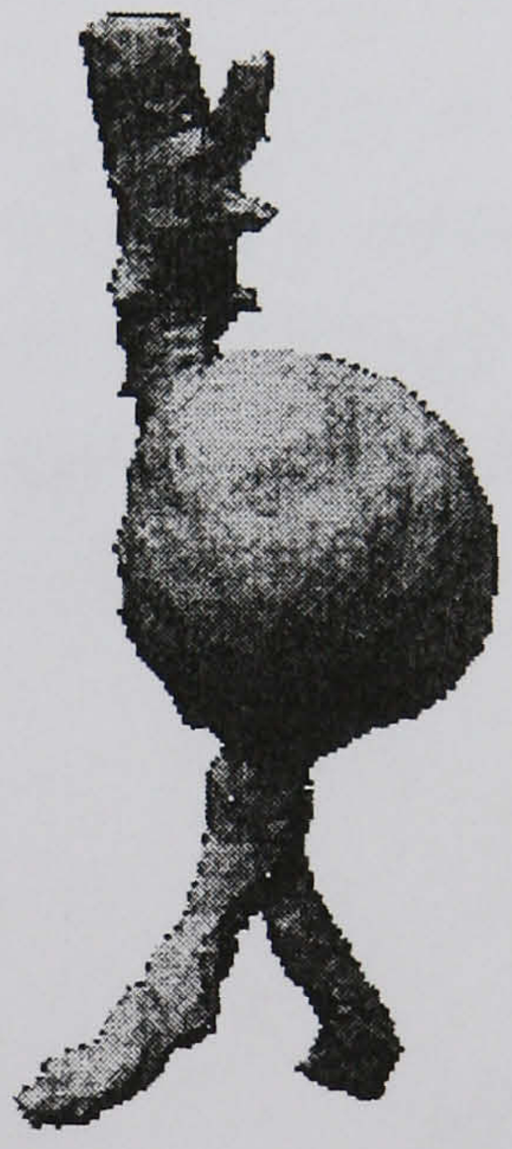
- Generic data can be used in training, however, patient-specific data provides a variation that is useful in training.
- Providing the physiology of the body- e.g. bleeding or breathing- is important for some medical simulators; however, it requires expensive modelling.
- Accuracy of the anatomical, physical and physiological model of the organs is essential; however, interaction is the trade off.
- The interface between the physician and the simulator should not form an extra burden.
- Force feedback is important in some surgical procedures- e.g. joint replacement and minimally invasive surgery.
- Evaluation of the performance is important in planning and training.
- Providing distributed education and training applications adds to the value of the application, since it allows for group learning.

Virtual reality is able to deliver most of these requirements. Visual realism of organs is possible by 3D rendering and stereoscopic viewing. Behaviour realism can be provided by modelling physical reactions of objects and simulating physiological events. The advantage of using virtual reality simulators is that it provides safe controlled medical environments to practise and prepare procedures. Different scenarios can be built in the simulators to provide a wide experience that can be repeated as needed. The medical simulator is a consistent method of training as well as a means of retaining the medical skills. In addition, evaluation capabilities can be added to the simulator.

However, current virtual reality technology suffers from some limitations. Virtual reality input devices are not accurate and fast enough for tracking. Delay between the user action and the environment reaction disturbs the perception of information. In addition, some peripheral devices are not comfortable to use. Because medical applications require high accuracy and are complex by nature, they demand high computational power. Furthermore, to achieve closed loop interaction, the processing cycle- i.e. detection, simulation, rendering- should be interactive. Hence, some systems trade off one aspect in return for the other; however, systems that choose not to compensate accuracy nor interaction are pushed into expensive solutions in terms of hardware and software. On the other hand, virtual reality can support group training via special peripheral devices; however, this training is limited to certain computer platforms and collaboration requires the participants to be at the same physical location.

Chapter 3

Motivating application: Abdominal aortic aneurysm



In the mid 1980's, a new surgical paradigm was introduced. This novel methodology aims at reducing the harm caused by surgery, hence increasing the chance of survival and reducing the recovery time. This methodology has acquired its name from this goal; so surgical methods which follow this paradigm have been termed "Minimally Invasive Surgery" (MIS). The idea of MIS is instead of cutting through healthy tissues and organs to reach the diseased area- as in traditional open surgery- a small incision in the body is used instead. Hence came the more colloquial name for this type of surgery, "key hole surgery". Minimally invasive surgery differs from traditional surgery in a number of ways- as shown in Table (3.1).

	Minimally invasive surgery	Traditional surgery
Access	small incision	expose the diseased area
Manipulation	long instruments	direct touch
Vision	2D video monitor	direct vision

Table 3.1: Differences between MIS and traditional surgery

MIS procedures are performed from a small incision usually far from the diseased area. Long instruments are used to gain access to the operating area. The physical cues during the operation are visual and haptic: a video monitor shows what is happening inside the body, while touch sensation comes via the long instrument. This contrasts with what surgeons are used to in open surgery, where they have full visual and touch access to the organ. Because of these differences, the MIS surgical methods require training which is different from the traditional techniques. Thus, in addition to learning the steps of the procedure, the surgeon should learn to coordinate his manipulation of the surgical tool based on the video monitor in front of him. The surgeon relies on his skill to mentally construct the actual three-dimensional structure of the anatomy and the tool from the two dimensional image on the monitor. Furthermore, the surgeon is trained to perceive the incoming force from the surgical instrument and interpret the current position of the tool based on this force. Figure (3.1) shows the special skills needed in MIS procedures.

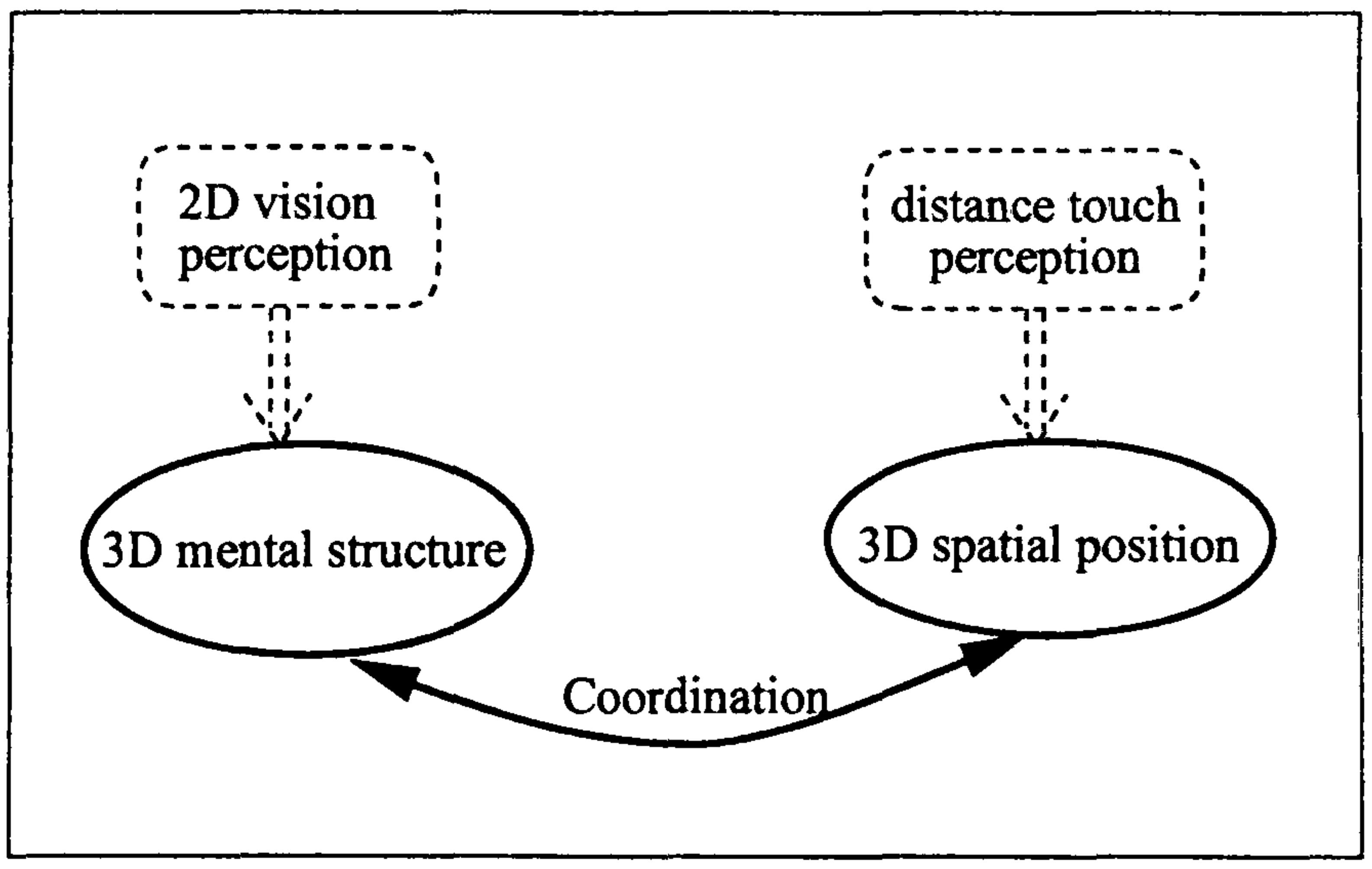


Figure 3.1: Skills needed for minimally invasive surgery

There are a number of surgical methods which are described under this general MIS term. Endoscopy, laparoscopy, and interventional radiology- to name a few- have the common paradigm of minimally invasive; however, each has its own characteristics. For instance, endoscopy is usually used for procedures which utilise a small camera mounted on the tip of a long instrument for diagnostic purposes. Laparoscopy uses a camera, as well as surgical instruments- such as grasper, clip applier, suturing device..etc.- mounted on long shafts to perform surgical procedures. Finally, interventional radiology uses imaging modalities to guide invasive diagnostic or therapeutic procedures. These different characteristics call for different training requirements. MIS is a suitable application for computer-based simulation since it is procedurally dependent and places the user at a distance from the operational area which facilitates the transition to and from the simulation environment [85]. In this research we focus on interventional radiology procedures. Two reasons supported this direction: one is the practical need expressed by radiologists in St. James's University Hospital in Leeds for new training methods for this type of procedure; the other is, the simplicity of this type of procedure compared to the other types which involve more complex instruments and manipulation. It thus makes it a suitable vehicle to study the questions underlying this research.

3.1 Interventional radiology

Interventional radiology utilises imaging modalities such as CT, MRI, ultrasound and fluoroscopy to guide the procedures. Interventional radiology is divided into two classes: angiographic procedures which take place in the vascular system, while soft tissue procedures take place in the urinary and bronchial systems. Both types utilise guide wires and catheters to perform the procedure. The catheter is a flexible tube for insertion into a narrow opening so that for example fluids may be introduced or removed [62]. Meanwhile, a guide wire is used to facilitate catheterisation of tortuous or atherosclerotic vessels, while reducing internal damage [146]. Both the guide wire and the catheter are long tubular devices with typical diameter between 0.24-0.81mm and length between 80-180cm. The guide wire and the catheter have different tip shapes which makes them function differently.

Interventional radiology like other forms of MIS requires special training to acquire the skills mentioned earlier. A practical training course in interventional radiology offered by M.D. Anderson Cancer Center in Texas can give an idea of the training methods utilised. According to Wright [189] the course provides the following:

- "Hands on" training on pigs to perform the procedures under supervision.
- "Hands on" bench-top practice with instruments from different vendors.
- Instructional video.

- Discussions about the specifications of the different products available in the market, when to use them and complications expected when using them.

In addition to such a course, the trainee would probably observe a number of procedures done on patients before they actually perform one. There are a number of difficulties with such training methods. For instance, not all countries permit the use of animals for training as well as this being expensive. Second, some training centres use plastic models, which are far from being a realistic environment for training. Relying on observation to establish the basic training is not sufficient. Consequently, trainees performing procedures on patients increase the risks involved in these procedures. Hence, frequent practice is what is required for training. This practice should be performed in a safe environment which mimics the actual anatomy and physiology of the body as closely as possible to ensure adequate transfer of skills. This is where computer technology can contribute by providing computer-based training systems- as shown in chapter (2).

The training course mentioned earlier was initially conducted in Texas. However, because of the popularity of this type of training, the course is offered in other centres- e.g. Norway, Belgium and Spain- at certain times [189]. In order to set up these courses in other centres a lot of equipment is needed as well as preparation effort. Trainees have to travel to the nearest centre to undertake the course. Thus, cost and effort are required from both the trainees and the training centre. An ubiquitous computer-based training is the answer to this problem.

In this research, we are investigating a computer-based training system that will address these issues, applied to an angiographic interventional radiology procedure which is used to treat abdominal aortic aneurysm (AAA). The rest of this chapter will discuss this particular procedure and analyse its training requirements.

3.2 Abdominal aortic aneurysm (AAA)

Abdominal aortic aneurysm is an abnormal ballooning or dilation of the abdominal part of the aorta- see figure (3.2). This disease is largely seen in males over 40 years old. On average 31.9 per 100,000 people develop abdominal aortic aneurysm (AAA). The normal diameter of the aortic artery is 2cm; in the case of AAA patients, an aneurysm of size 5-6cm requires immediate repair. This is due to the fact that as the size of the aneurysm increases, the risk of its rupture increases. In the case of a rupture, internal bleeding follows and the chance of survival is 50%. Another complication in AAA is the aortic dissection, where the arterial wall tears and blood leaks to the wall. Insufficient circulation and irreversible damage to the kidneys are among the possible complications. AAA can be detected using any of the following imaging modalities: abdominal X-ray, CT scan, MRI, ultrasound or aortic angiography. If the size of the aneurysm is larger than 4 cm or the rate of aneurysm expansion is greater than 0.5cm/year, surgical treatment is carried out [1, 168].

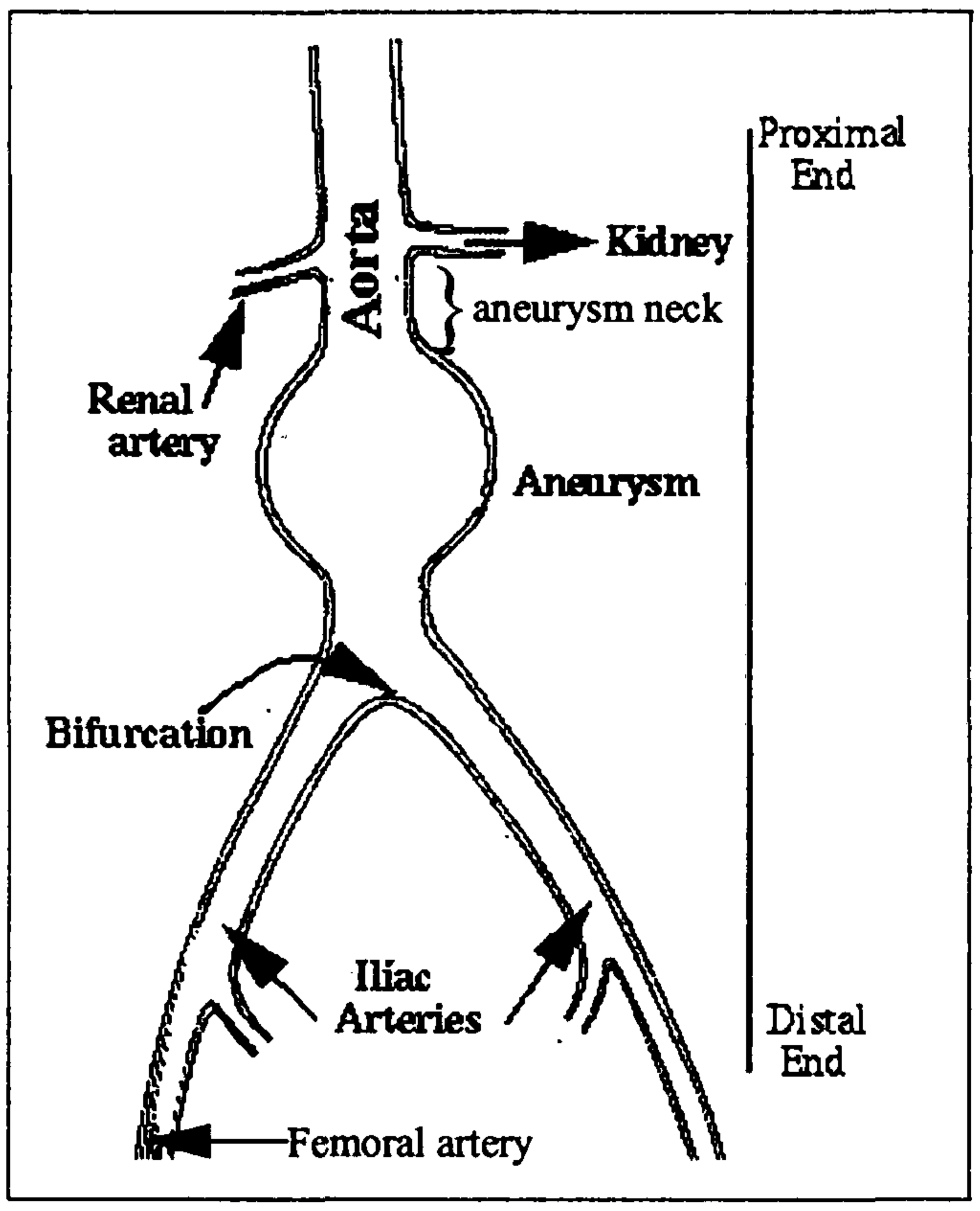


Figure 3.2: AAA anatomy

Traditional open surgery for this disease involves exposing the aneurysm, removing the calcification then placing a graft inside the artery. The mortality rate of this procedure is 1-4%, average stay in the hospital is 7-10 days (mostly in ICU) and the recovery time is 6-8 weeks [168, 40].

Interventional radiology has been used to provide a safer, less traumatic and less painful treatment. With an endovascular procedure no general anaesthesia is required, and the average stay in the hospital is 3 days. This reduction in patient instay results in a cost reduction from the point of view of health care and fast recovery from the patient's view.

The goal of the procedure is to place a stent-graft across the the aneurysm to exclude it from the blood circulation. The stent is a metal mesh (8-14 cm long and 18-25 mm in diameter) covered with a polyester fabric, which is the graft. The graft is fixed to the stent with sutures in the proximal and distal ends- see Figure (3.3). Furthermore, the stent-graft is part of a delivery system which consists of a pusher catheter, balloon catheter and sheath- see Figure (3.4).

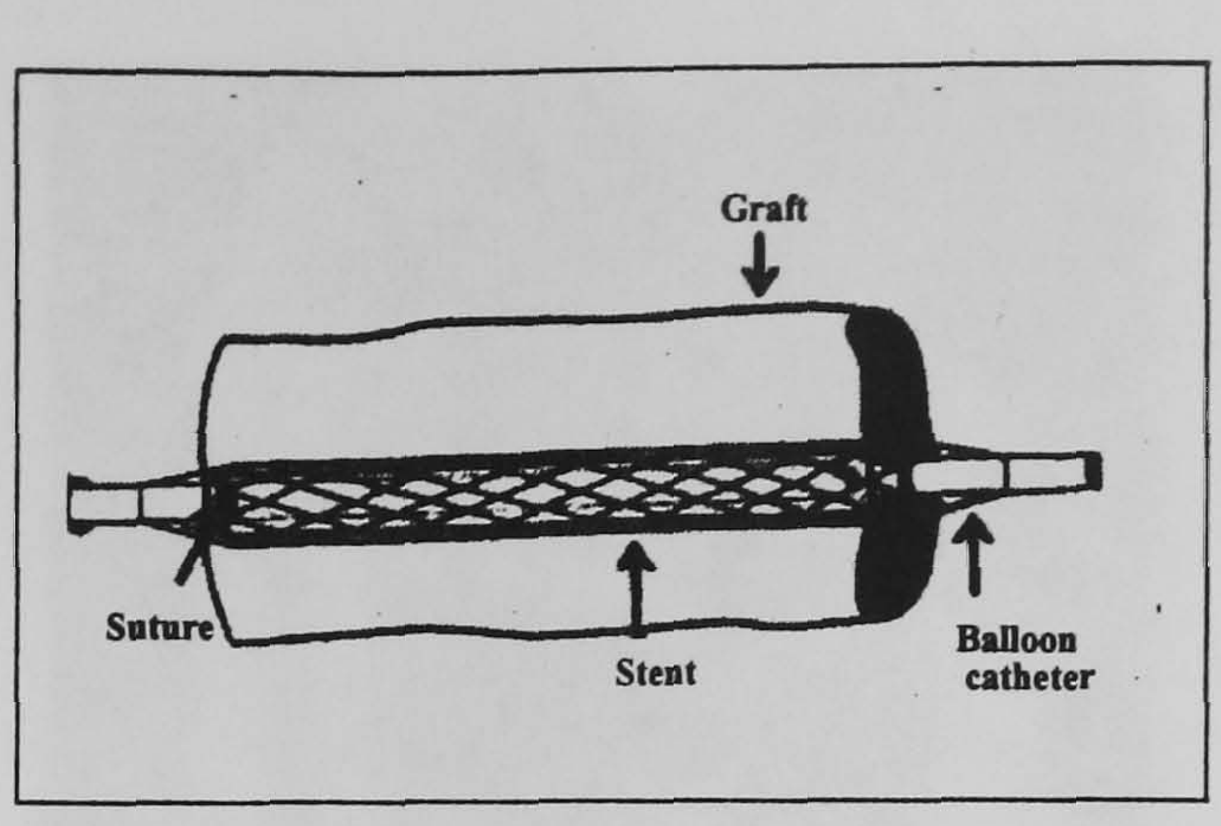


Figure 3.3: Stent-graft structure

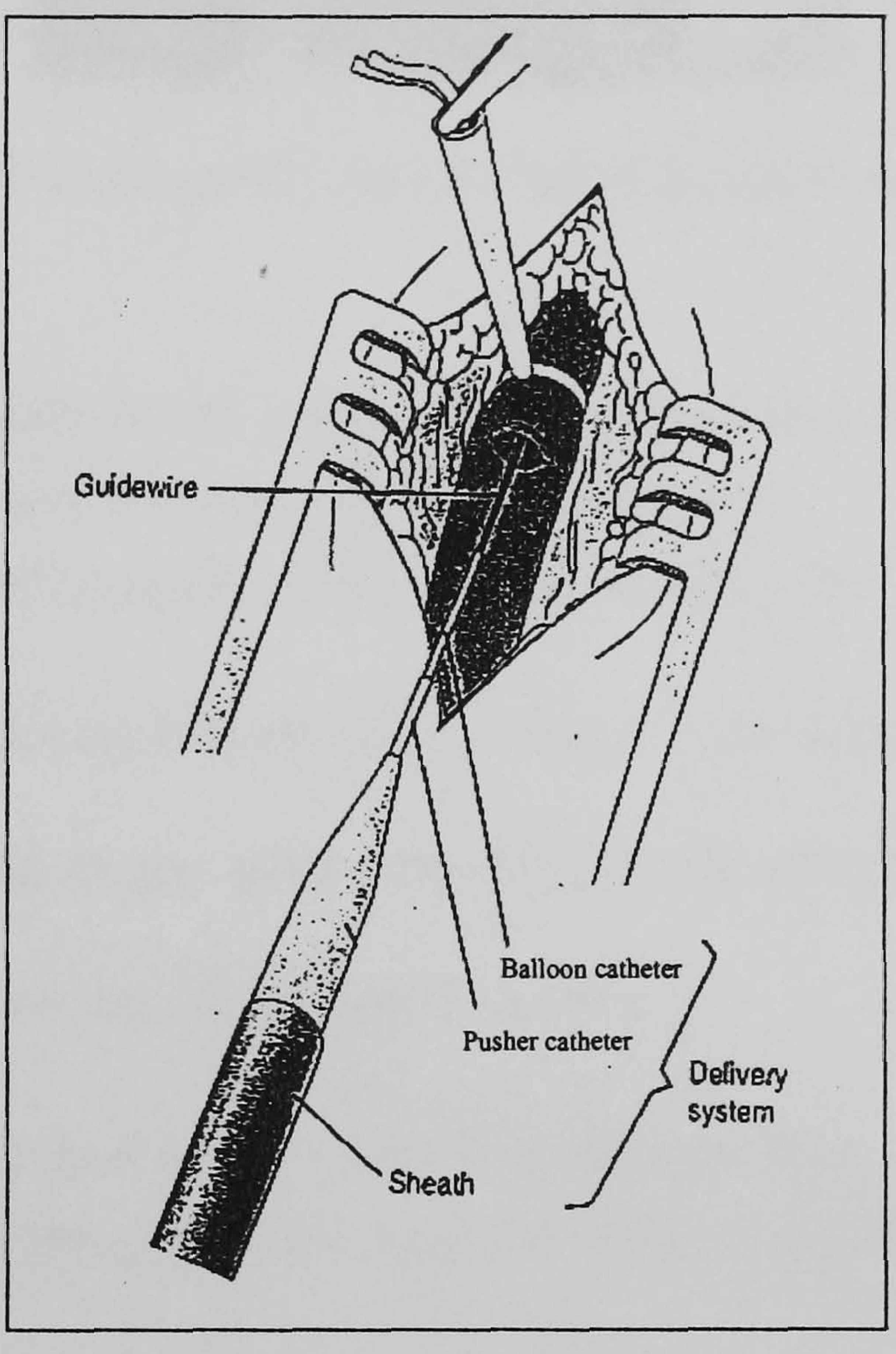


Figure 3.4: Stent-graft structure

This procedure is characterised by one of two modes, which depends on the position and extent of the aneurysm. A straight stent-graft placement is done when the aneurysm starts below the renal arteries and extends up to the bifurcation. On the other hand, a bifurcated stent-graft placement is performed when the aneurysm extends to the iliac arteries. Figure (3.5) shows the different types of stent-graft (image extracted from [40]).

To perform the operation, a number of measurements are required to verify the suitability of the case for this procedure. The distance from the renal artery to the aneurysm- known as *aneurysm neck* [refer to Figure (3.2)]- should be greater than 1 cm. The distance from the bifurcation to the distal end of the aneurysm, as well

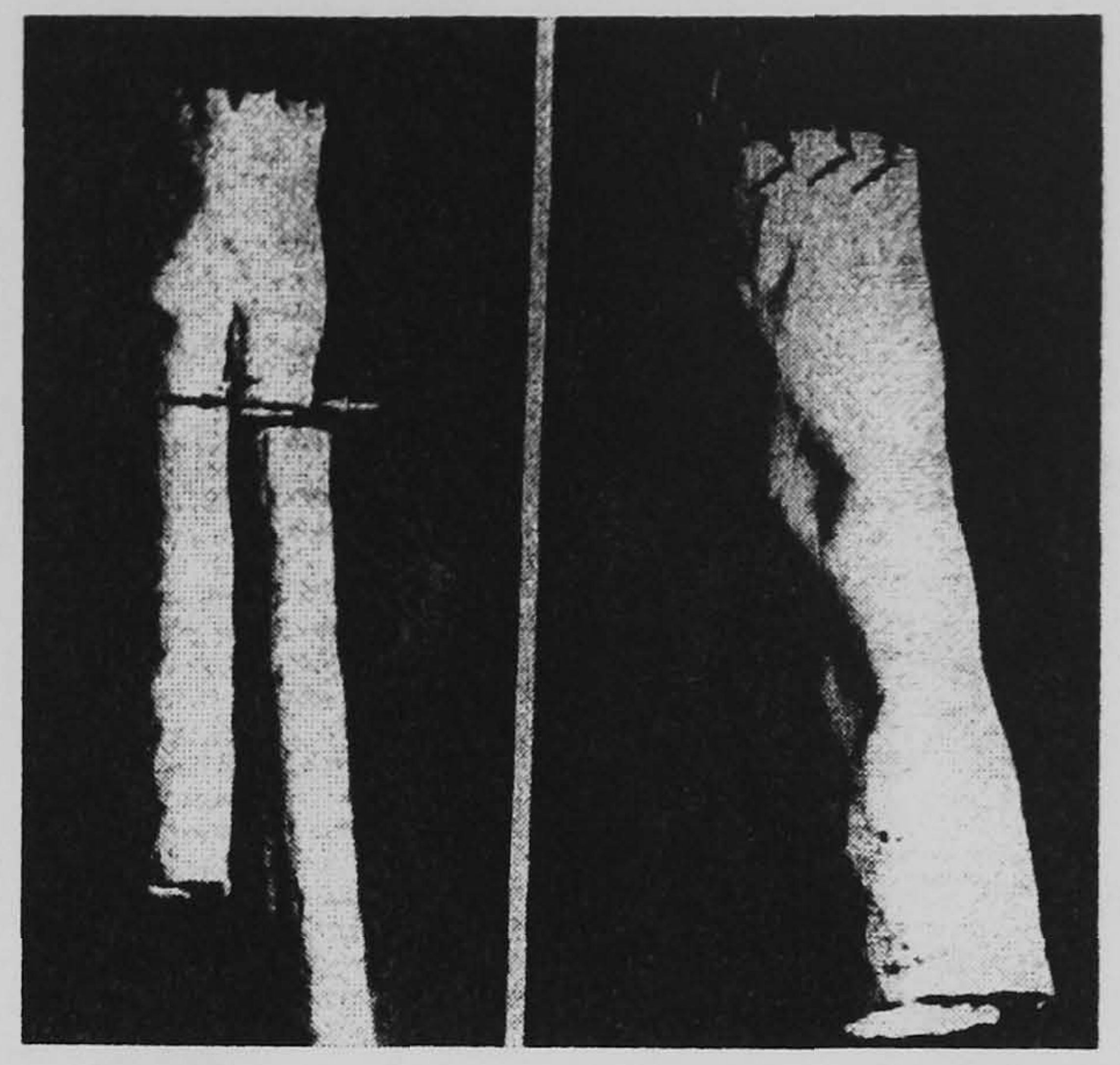


Figure 3.5: Two types of stent-graft: on the left bifurcated stent-graft, on the right straight stent-graft

as the diameter and length of the aneurysm are important pieces of information to determine the required size and length of the stent-graft.

The successful completion of the procedure is based on the following three criteria:

- Place the stent-graft such that the aneurysm is excluded from the circulation.
- No leaking of blood to the aneurysm around the stent-graft.
- The stent-graft does not block any branches.

Figure (3.6) shows the goal of the endovascular operation. In the next section we will discuss the steps of the procedure and the skills required for each step.

3.3 AAA: Task analysis

When developing a simulation of a human-machine system, it is useful to carry out a task analysis which will identify and describe units of work which contribute to the accomplishment of the system goal. In training applications, task analysis is useful since it allows the identification of skills and knowledge needed for each function in the system, helping in its design and evaluation [95, 57]. To perform task analysis of the AAA stent-graft deployment procedure we have made use of the following sources of information:

- Endoprosthesis training manual (provided by radiologists in St. James's University Hospital).

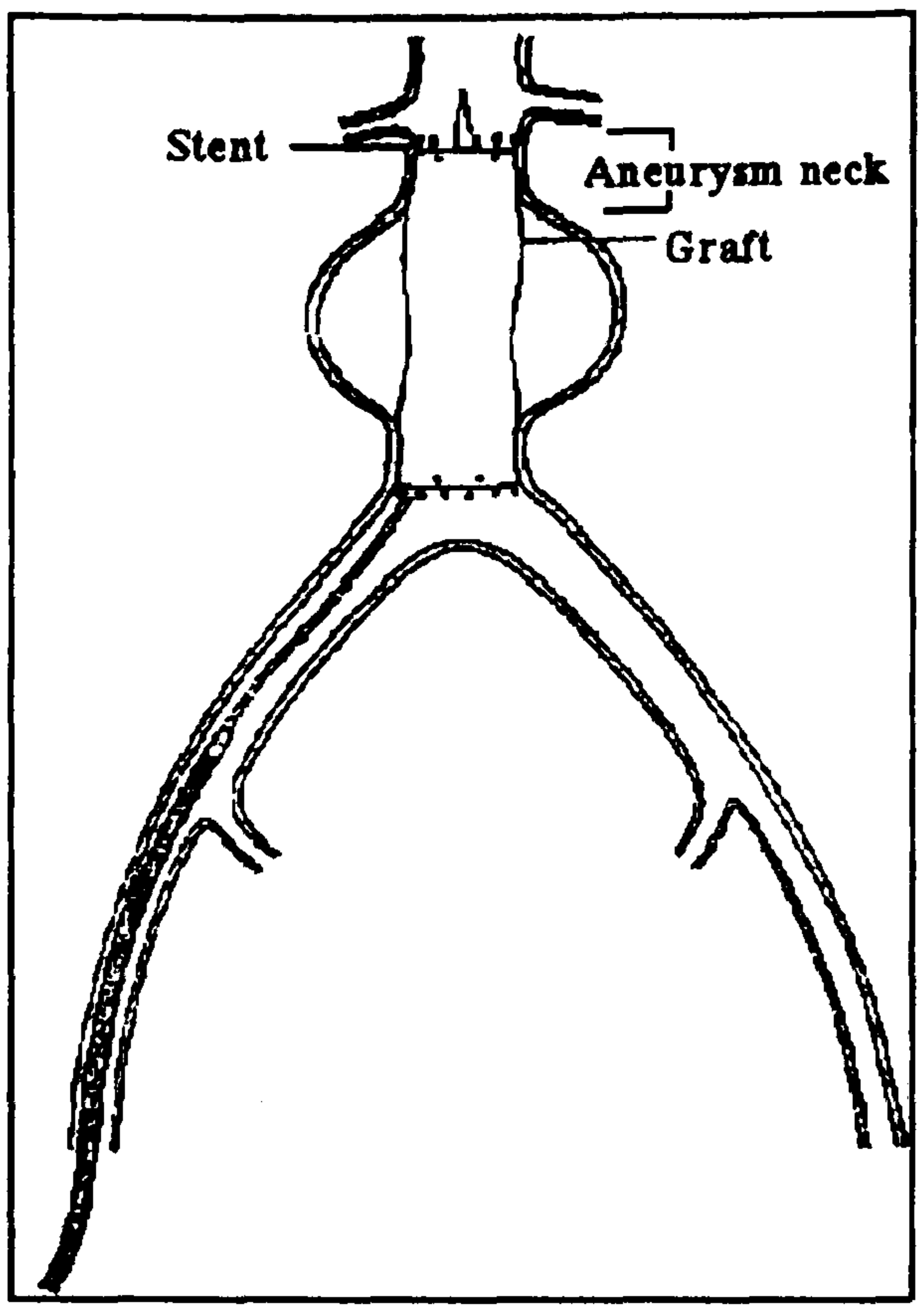


Figure 3.6: stent-graft placed across the aneurysm

- Observation of renal plasty procedure.
- Conversation with radiologists and surgeons doing this type of operation.
- Practical experimentation on a physical mock up of an aorta with AAA and a set of guide wires and catheters provided by St. James's University Hospital.

We will use the hierarchical task analysis method to present the functions involved in a straight stent-graft deployment procedure. In this method, we start from the general goal of the system. The procedure is then divided into tasks, which are divided further into sub-tasks. While doing that, we will classify each task according to the type of skill required to perform it. The skills are categorised according to the amount of mental activity required [155]. These are, in descending order of mental activity:

- Decision making skill: A complex cognitive activity which involves choosing the most adequate among alternatives.
- Representational skill: A mental representation of a process or object which improves the performance of the task.
- Procedural skill: A sequence of cognitive and physical activities performed in predictable situations.

- Automated skill: A cognitive or physical activity which is performed rapidly, accurately and with minimum mental processing.
- Psychomotor skill: A skill where movement is based on kinesthetic cues.

This classification of tasks will determine the type of training needed to transfer it to the desired environment. For instance, automated skills are better transferred by observing novice users and practising the task. This is due to the fact that experts make no effort in such tasks so that they are carried out subconsciously. Procedural tasks are transferred by giving the basic knowledge first using verbal methods, then compounding it with practice which gives proper feedback to the user actions. Knowledge of any representational skills needed in the operation will enable the tracking of errors during the operation to faulty mental models. This type of skill is best transferred in a media which represents the model pictorially to the trainee. Decision making skills are developed by experiencing diverse situations. Hence, a media which will offer a range of problems and provide the trainee with direct response to the decision is a training requirement [155]. This analysis of skills and knowledge will assist in concluding the necessary requirements of a training system for our procedure.

Figure (3.7) shows the first level of the hierarchical task analysis of the procedure. The endovascular stent-graft deployment is divided into four main tasks. The first task-Task 1- involves the access of the artery where the tools will be inserted and manipulated. Images and measurements required for this procedure are carried out in this task.

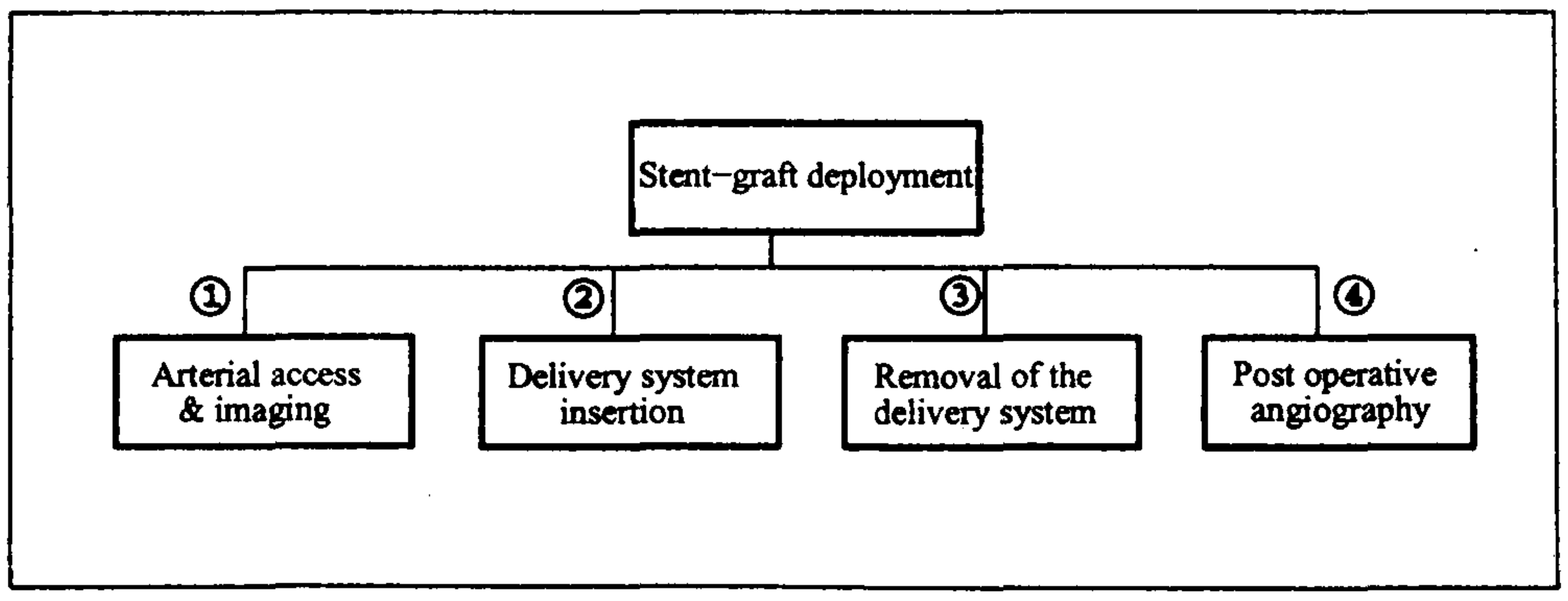


Figure 3.7: First level classification of endovascular procedure

In the second task- Task 2- the delivery system is inserted and the stent-graft is deployed. This task is the most critical task of the procedure. The third task- Task 3- is to withdraw the delivery system. Finally, post-operative angiography is carried out to verify the success of the deployment of the stent- Task 4.

In task 1, the surgeon exposes the right femoral artery- refer to figure (3.2)- with an incision 5-7 cm long in the skin, and then incises the femoral artery to insert the tools. The angiographic catheter is inserted from the same incision, the left side of the femoral artery or the arm. During the insertion of the angiographic catheter some difficulties might be experienced due to a tortuous iliac artery. In this case, it is recommended to use a combination of shaped catheters and hydrophilic guide wires to facilitate access. An angiographic image is taken, then a contrast material is injected and the imaging is repeated. A digital subtracted image of these images appears on the monitor, showing the vascular structure. The image intensifier is positioned such that both the renal artery and bifurcation appear on the monitor- a field of 22cm is needed to cover both. The radiologist marks the position of the renal artery and bifurcation with a tape on the screen. After this is done, the image intensifier must not be moved. A marked catheter is then inserted to measure the aneurysm neck and the distance from the bifurcation to the aneurysm. Finally, a J-shaped stiff guide wire is inserted into the right femoral artery to pass the delivery system over it. Radiologists try to limit the number of times angiography is performed during the procedure, because of the dangerous effect of the X-ray on the patient and on the medical staff. This explains why insertion of the guide wire is done almost blindly. Force felt at the end of the guide wire is used as a cue to the position of the tool in relation to the structure. Figure (3.8) shows the sub-tasks involved in Task 1. A coloured dot is used to classify each sub-task according to the previously mentioned skill type.

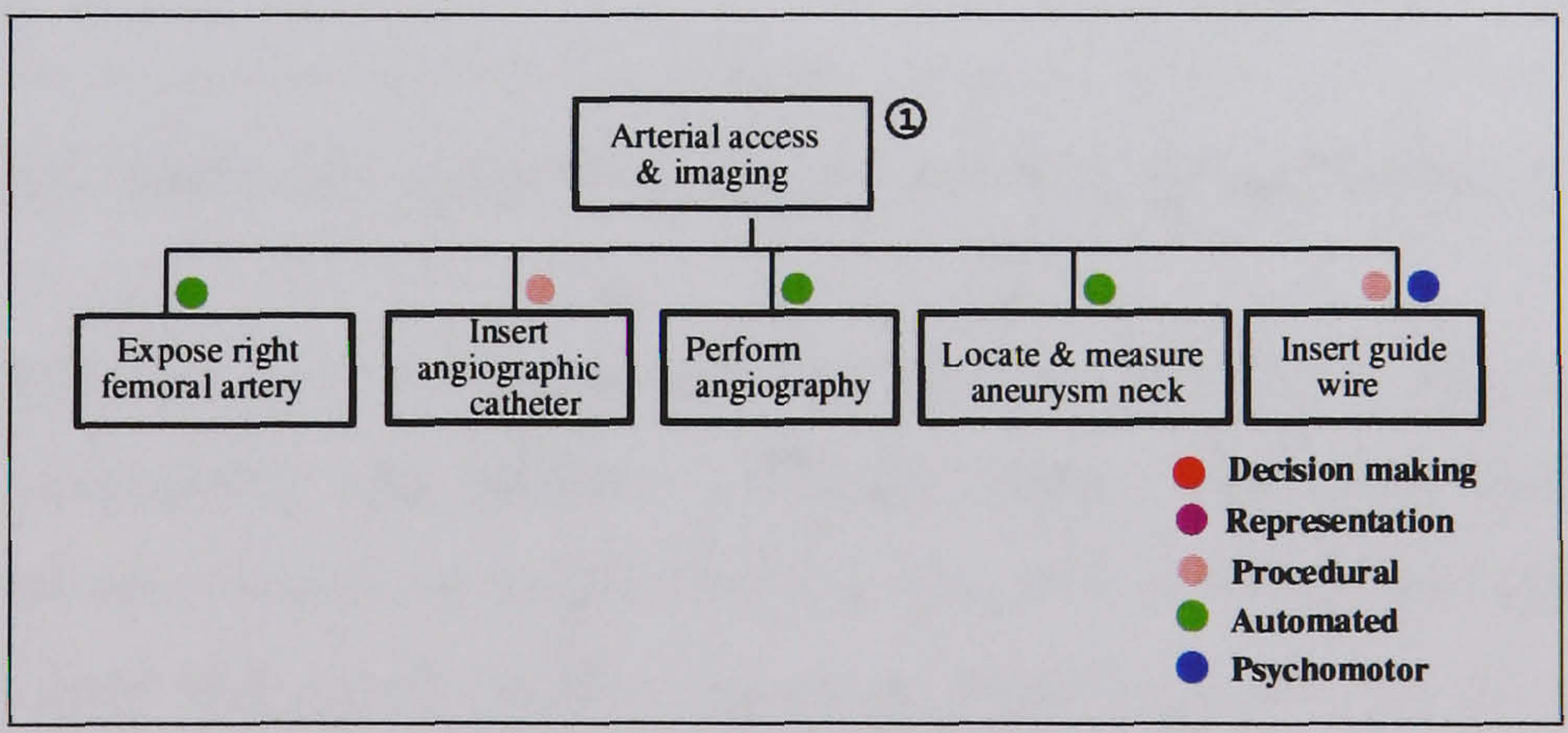


Figure 3.8: Subtasks involved in arterial access

The first sub-task in task 2, is to pass the delivery system over the guide wire. Such a task is not always a straightforward one. When the iliac artery is tortuous, the delivery system would need some manipulation- such as rotation- to avoid the angulation in the arteries. The arteries can be straightened by palpating the iliac artery or applying pressure on the abdomen. If it is impossible to avoid the tortuous

LEEDS UNIVERSITY LIBRARY

structures, the left side arteries may be used instead. The second sub-task is to advance the delivery system until it reaches the renal arteries. Angiographic images are repeated here to make sure of the position of the tool and to check that the distal suture is above the bifurcation- i.e. the length of the stent is correct. The stent is usually positioned 1-2 mm across the renal artery, that is because the end position of the stent is usually 3-4 mm lower than at the start of the deployment. Then the sheath is unlocked and withdrawn to deploy the upper stent. Caution should be taken during this task not to move the stent while pulling the sheath. When the aneurysm neck is not straight, it is difficult to judge the correct position of the upper stent. In this case, angiographic imaging should be taken from a perpendicular position if possible. Finally, the radiologist checks the position of the downstream stent and withdraws the rest of the sheath to deploy it. If the downstream stent needs shortening, the upstream stent is released using the pusher and the balloon catheter and the stent length is adjusted. Figure (3.9) shows the sub-tasks of task 2 and their classification.

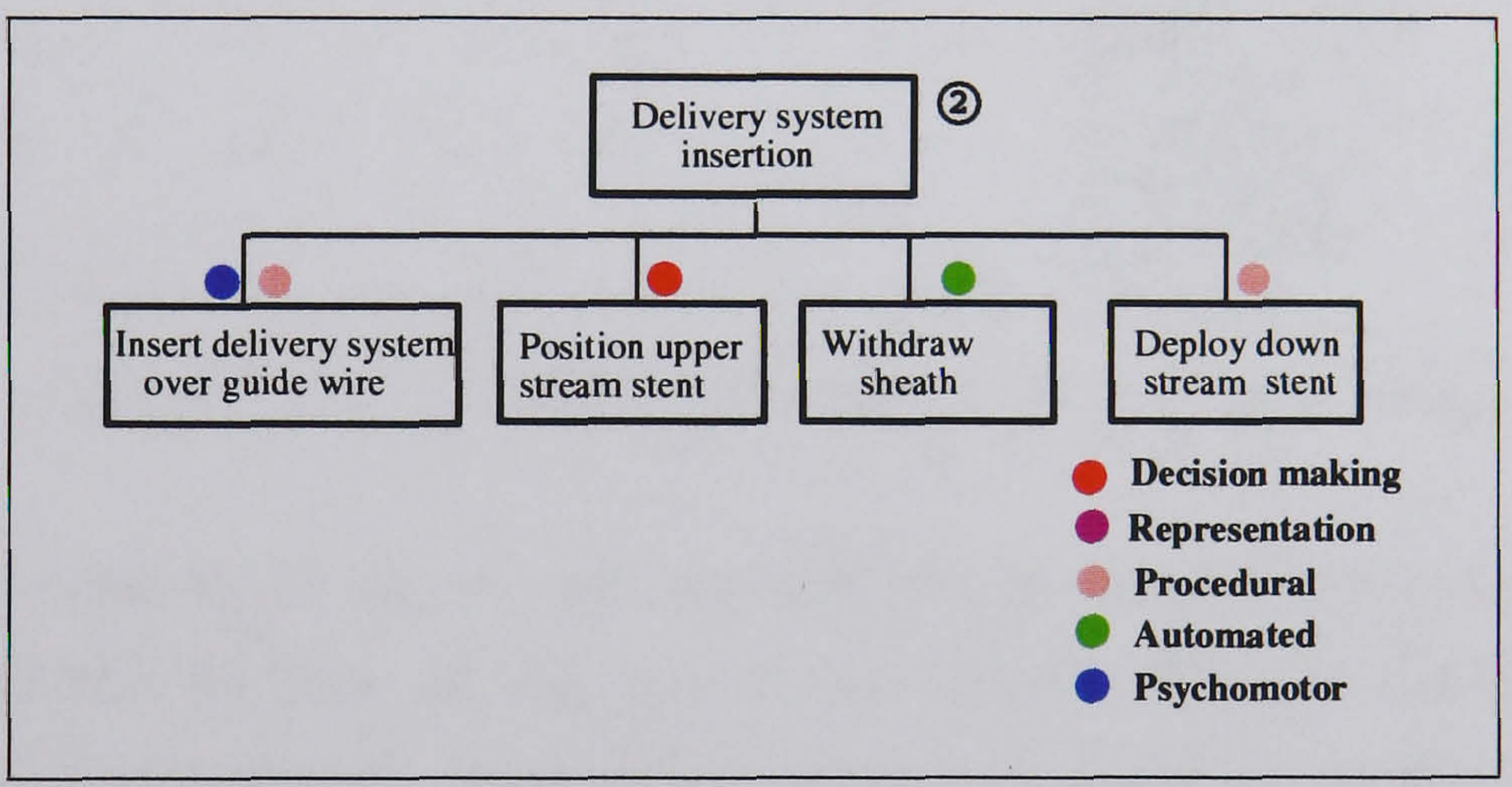


Figure 3.9: Subtasks involved in the insertion of the delivery system

Figure (3.10) shows the sub-tasks of task 3. In this task the delivery system is removed by retreating the balloon catheter first. Then the delivery system is withdrawn. Caution should be taken during the withdrawal operation, so that the delivery system does not catch on the upper or lower sutures, which causes the stent to migrate. If difficulty is encountered when withdrawing the tool, angiographic imaging should be performed and the sutures are exposed to release the delivery system.

Finally, angiographic images are taken again to check the position of the stent- see figure (3.11) for subtasks of Task 4. The angiogram will show if blood leaks from the distal or proximal end as well as the reason for the leak. Table (3.2) shows the possible complications and the required remedy.

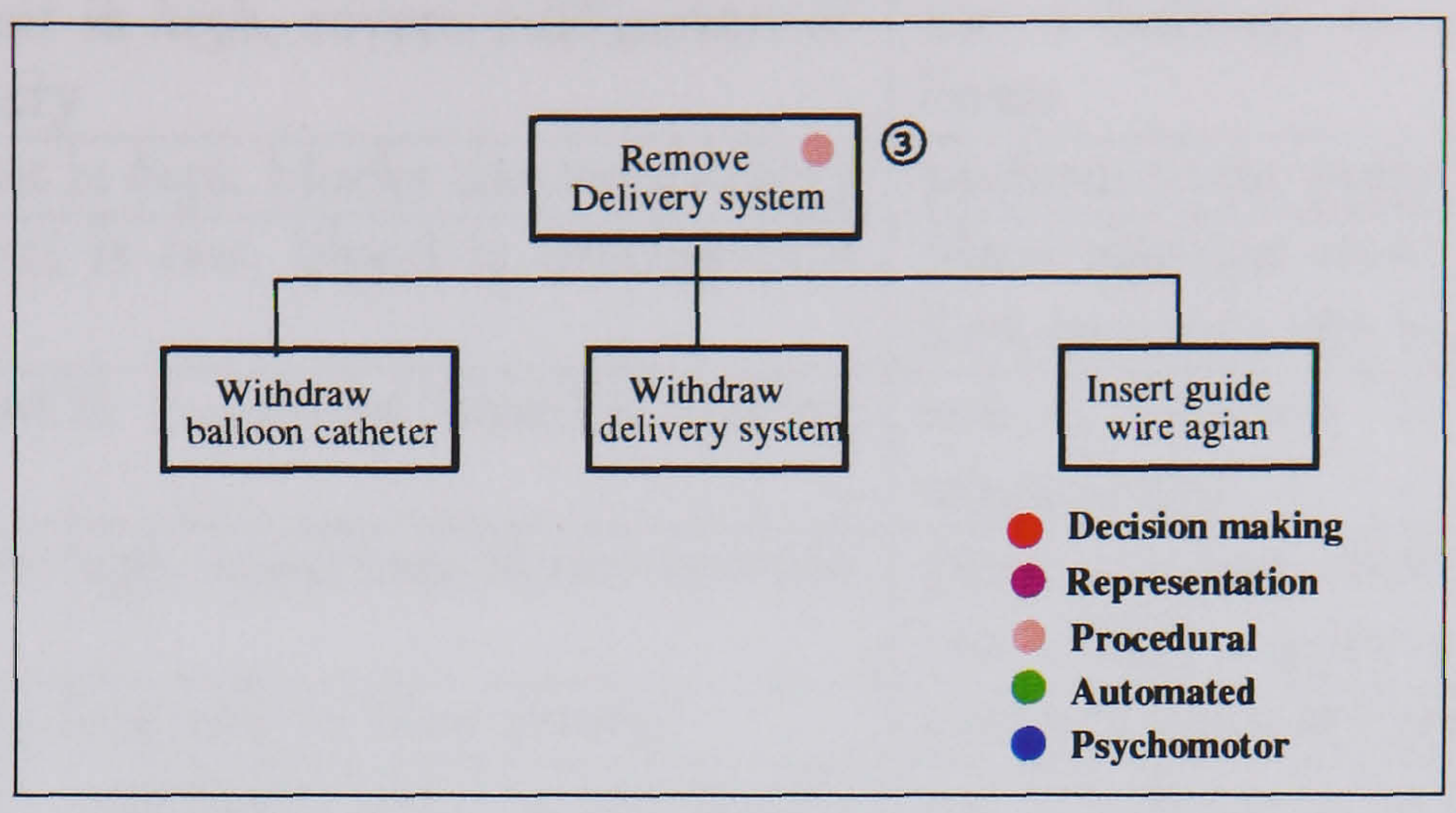


Figure 3.10: Subtasks involved in removing the delivery system

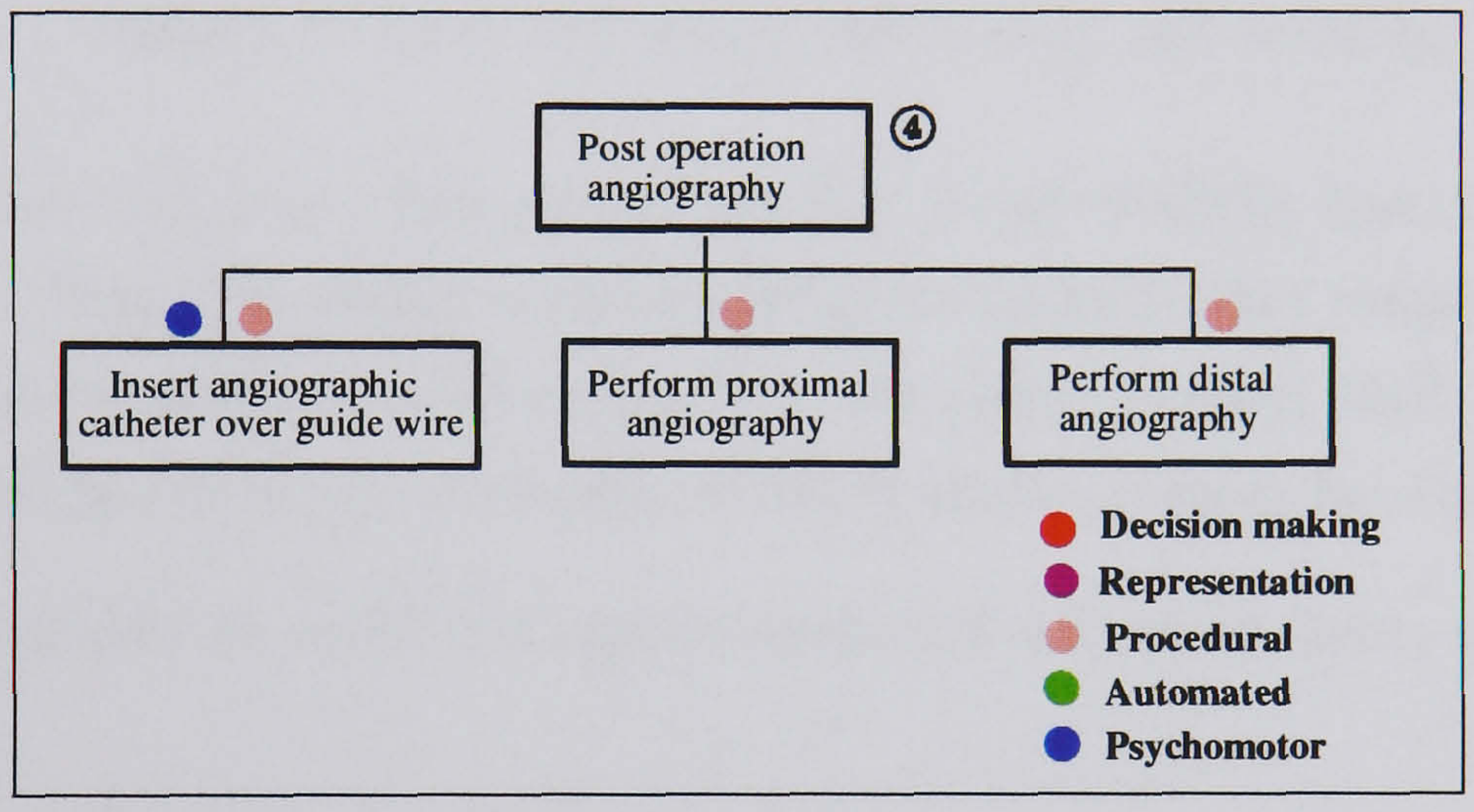


Figure 3.11: Subtasks involved in post angiography

From the previous analysis, we can see that there are two critical pieces of knowledge that affect the success of the operation. First, there is the 3D structure of the anatomy of the patient. Knowledge of any angulation in the aneurysm neck and complex twisting of the aorta will affect the positioning of the stent and the manipulation of the tools. This type of knowledge is based on a representational skill where the 2D pre-operative and intra-operative images are constructed into 3D mental representation of the anatomy. Second, the initial measurement performed in Task 1 will affect the choice of the stent used in the procedure. Inadequate choice of stent- i.e. shorter or longer stent- would require extra work to mend the problem. Although this task depends on automated skill, it is affected by the previous representational skill. The adequate positioning of the stent requires a decision making skill which is based on the previous two pieces of knowledge. Furthermore, the procedure involves three special skills: manoeuvring the tools, constant eye/hand coordination and force perception.

Proximal stent is *high*, covers millimeters of the renal artery	use a balloon to pull the stent down
Proximal stent is *high*, blocks the renal artery	perform open surgery
Proximal stent is *low*, blood is leaking into aneurysm	place another stent on top of the first to cover the leak
Proximal stent is *angulated*, blood is leaking	use a balloon to correct the orientation.
Distal stent is *high*, aneurysm is not covered	place another stent to cover the rest of the aneurysm
Distal stent is *low*, covers iliac artery	perform open surgery
Distal stent is *angulated*, causing retrograde leak	use a balloon to seal it.

Table 3.2: Incorrect stent positioning and remedy

It is clear that this procedure covers the full range of skills from motor to highly cognitive ones. Hence, in order to facilitate the transfer of this range of skills, multimedia needs to be utilised to accommodate the needs of each skill, as will be clear from the following list of requirements of the training system for this procedure:

- Pictorial media to build the *representational* skill of anatomy modelling in 3D space.
- Tactile media which gives force feedback in response to user actions to build the force perception- *psychomotor* skill.
- Verbal instructions and practical environment to practise the *procedural and automated* tasks such as the measurement task.
- An environment with a wide range of anatomical models to practise the *decision making* process of positioning the stent. The environment should simulate the effect of the decision- e.g. arterial block, leaking aneurysm or successful procedure.
- An interactive environment for the manipulation of surgical tools and getting the result of this manipulation such that the visual/ kinesthetic interaction is similar to the required *eye/hand coordination* in the real life situation.

3.4 Proposed solution (WebSTer)

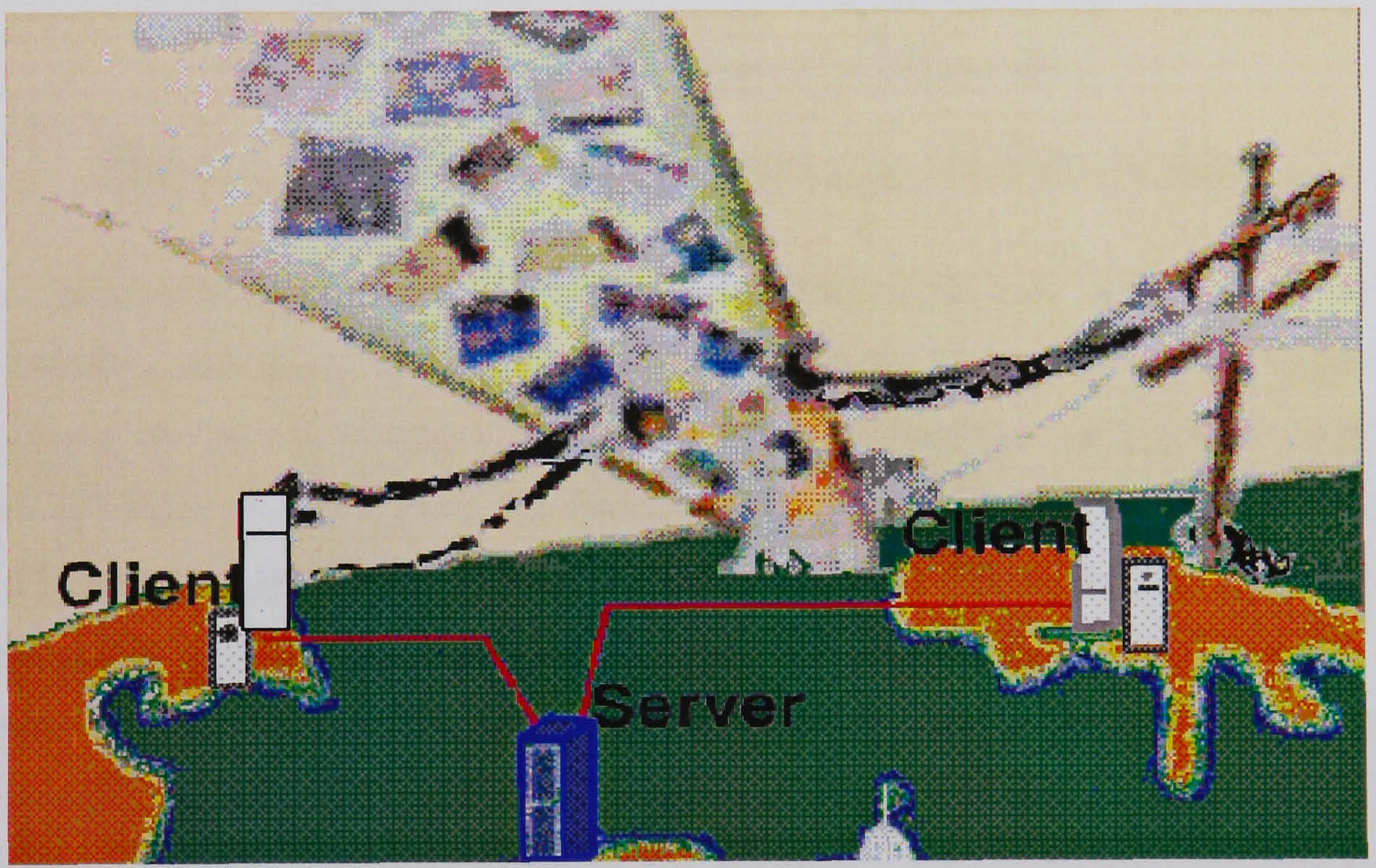
In section (3.1) we have highlighted some problems in the current training methods and concluded that a computer-based solution might be the answer. We have also recognised two important features in this solution:

- A realistic controlled training that targets certain surgical skills.
- An accessible affordable training.

In this work, we attempt to provide such a training system for interventional radiology procedures- with AAA being a case study. We study the potential of virtual reality and simulation technologies to provide the first aspect and the World Wide Web to be an environment that allows the second. Because the presented solution incorporates different disciplines, we have divided our presentation into three main components: the environment, the interface and the modelling. The environment-chapter (4)- represents the web society, where we will provide our solution. Under this component, we will review the current state of the web, the technologies involved, the requirements of the web, issues of our training which are concerned with the web and finally evaluation of the component. The interface component- chapter (5)- will present different virtual reality interaction techniques and the interaction requirements of our training system. Then, we describe our training system interface and evaluate this component. Finally, the modelling component- chapter (6)- will present background issues of modelling behaviour, list the requirements of this component, describe the techniques that we have used in modelling the surgical training system and evaluate these techniques. Hence, each component of our system will give a background of the involved technology, then list the set of requirements of this component for our application, describe our implementation and finally evaluate this implementation. The reason for such a structure is that each component represents the contribution of an independent field with special requirements. Hence, the evaluation at the end of each component will discuss how much of these requirements were achieved and limitations that hindered other requirements. Meanwhile, chapter (7) will evaluate the overall result to see how useful it is for our surgical training purpose.

Chapter 4

WebSTer environment



4.1 Introduction

In 1987 Kobayashi reported his vision of a "Man and Computer & Communication society" [106]. He anticipated that humans will be interacting within a computer and communication (C&C) society by the end of the century. The aim of this society is to meet individual needs both on the business and personal levels via a group of applications which cover different fields of interest. The World Wide Web is the embodiment of this vision in our present lives. The Internet forms the environment which hosts this web society. Until recently, the web society has essentially taken the form of a huge database of all sorts of information - fulfilling one aspect of user needs. Today, however, we see other individual needs being targeted and explored, as the web turns into a distributed computing environment. During the last two years, attention has been directed to provide applications and tools for different fields on the web. Hence, recently the web society has been reforming its shape to look more like the C&C society that Kobayashi anticipated.

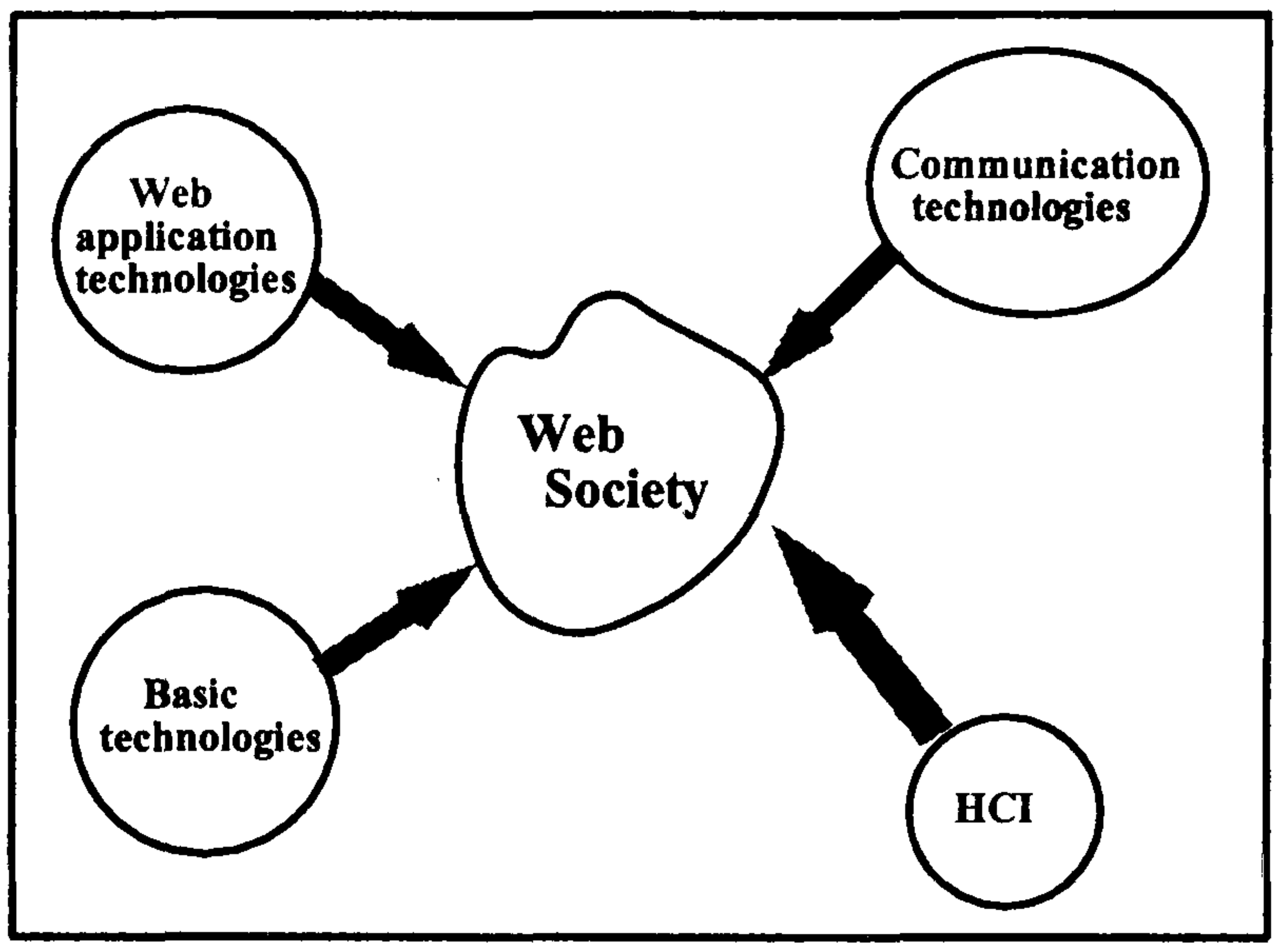


Figure 4.1: Technologies contributing to the web society

In this chapter we will look at this web society and how we utilised it in our surgical training solution. In section (4.2) we will set the scene by presenting networking concepts which support the web. Then in section (4.3.1), we will define web-based applications and describe the different architecture designs of such applications. There are a number of technologies that support web applications. These technologies can be classified into four categories, as shown in Figure (4.1). *Basic technologies* are fundamentals that existed long before the web society and are being migrated to this new environment, such as simulation, computer graphics, scientific visualization,.. etc. The *communication technologies* are protocols, middleware and software that have emerged to facilitate communication between web applications-section (4.3.2) will give some examples of these technologies. *Web-application technologies* refer to languages and tools that help in building applications within the web society; section (4.3.3) will review some of the popular technologies under this class. Finally, the *Human-Computer Interaction* (HCI) technologies are separated from the basic technologies although they could fit under this class, because the web society will require new HCI paradigms.

Figure (4.2) shows some examples of web-based applications that can be found currently in the web environment. These applications cover a variety of fields and

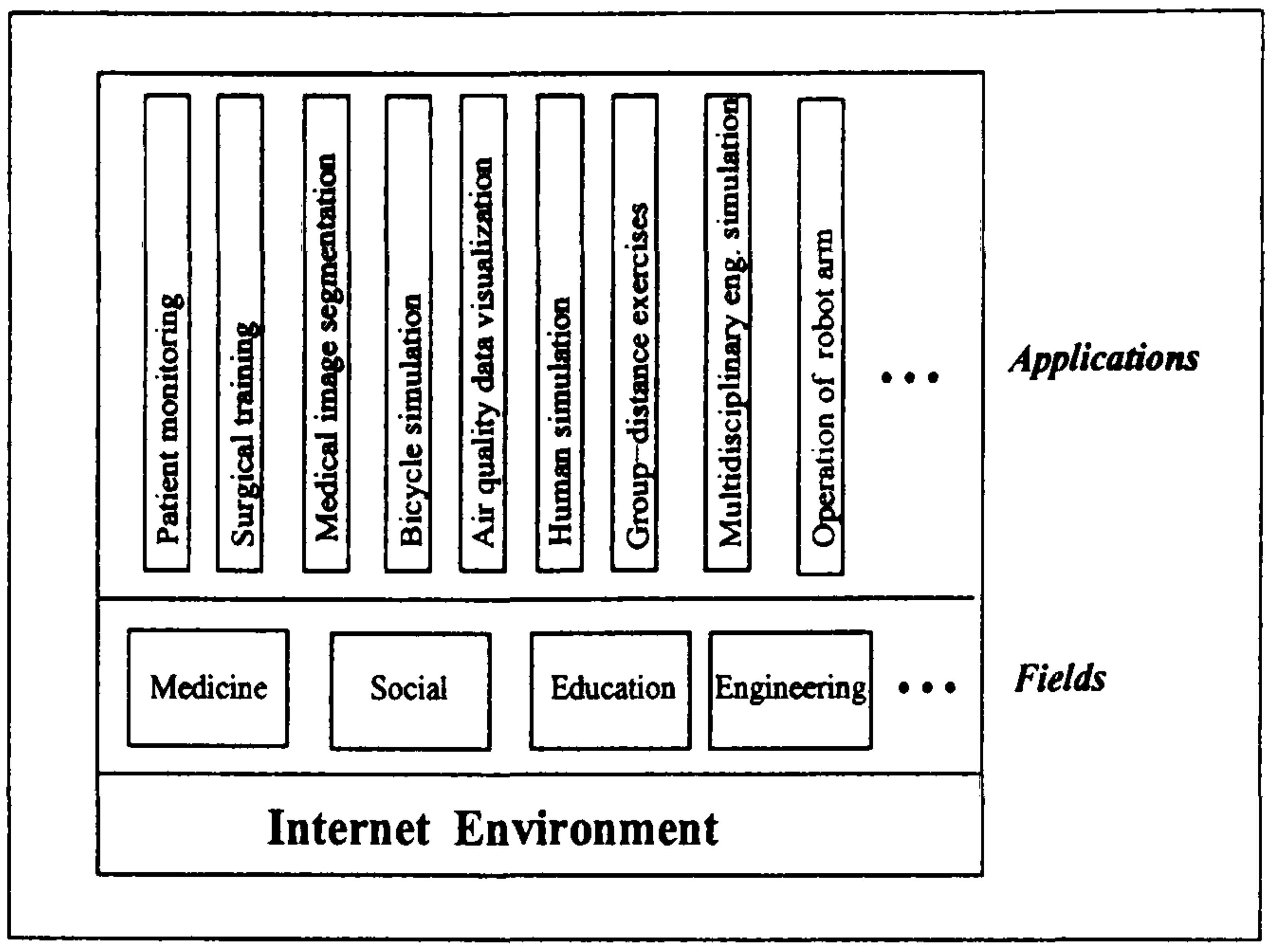


Figure 4.2: The web society

we will explore them in more detail in section (4.4). Then in section (4.5) we will establish the requirements of our surgical training solution from the point of the view of the web environment. Section (4.6) contains a description of our web solution, while section (4.7) reflects on this solution from the point of view of previous requirements.

4.2 Networking concepts

The Internet consists of world-wide heterogeneous networks that are different in their capacity and characteristics. These networks vary in the protocols they adopt to transmit the data. The Internet is largely based on the TCP/IP protocol. The Internet Protocol (IP) can run over any transmission media, hence facilitating communication between heterogeneous platforms. This is due to the fact that IP provides intelligent parts at the ends of the communication channel, while the network core is kept simple. However, because of this simple architecture, the IP protocol may cause delay and data loss when network congestion occurs at one part of the connection. Hence, the IP protocol is described as providing "best effort" service.

Under the IP network model, the next layer in the network architecture is implemented as either Transport Control Protocol or User Datagram Protocol. *Transport Control Protocol (TCP)* is a connection-based protocol that provides a reliable flow of data between two computers. Applications such as Hypertext Transfer Protocol (HTTP) and File Transfer Protocol (FTP) use the TCP connection since reliability and order of data transfer is important for these applications. On the other hand, *User Datagram Protocol (UDP)* is a protocol that sends datagrams from one computer to another with no guarantees on their arrival. A datagram is an independent, self-contained message; hence the order of delivery is not important. Some applications- such as a clock server which sends time stamps to a set of clients- do not require reliable communication (as occasional lost records are not crucial) but cannot tolerate delay caused by the overhead of guaranteed delivery. In this case a UDP is the best choice of communication.

The best effort service provided by the IP model is not satisfactory in some cases such as network telephony. The Asynchronous Transfer Mode (ATM) supports the needs of such applications. ATM is based on the principle of quality of service (QoS). Quality of service is defined as "the ability of a network element- e.g. application or host or route- to have some level of assurance that its traffic and service requirement can be satisfied". ATM achieves this by establishing a virtual circuit- a unidirectional connection between the source and the destination- and allocating resources to it according to the QoS required. ATM is able to satisfy the requirements of real time applications; however in a typical connection between two machines at least part of the network will be IP driven.

When an application is distributed over a network, the requirements of this application dictate the preferred type of the network to be used. In [19] applications are classified from the point of view of networking according to the rate of data transmitted and the degree of tolerance of delay. According to the first criteria, applications are characterised as stream data, where a relatively constant bit rate is transferred- e.g. video and audio- or burst data, where variable unpredictable bit rate of data is transmitted- e.g. FTP. On the other hand, the second criteria classifies applications as asynchronous, synchronous, interactive, isochronous or mission-critical applications. This classification describes the degree that transmission delay affects the functionality of the application in an ascending fashion. It is reported that when this feature becomes more critical in an application, the networking requirement is forced towards a QoS network support- i.e. ATM or bandwidth managed IP. In section (4.5) we will classify our training application according to these classes. Refer to [169] for more details about networking.

The World Wide Web (WWW) is an architectural framework for accessing linked documents on the Internet. Hyper text transfer protocol (HTTP) is the standard protocol used to transfer these documents. This protocol is used to send a request for a document on some host- which is called a server. The data is then downloaded to the host that made the request- called a client. Figure (4.3) shows a typical HTTP connection.

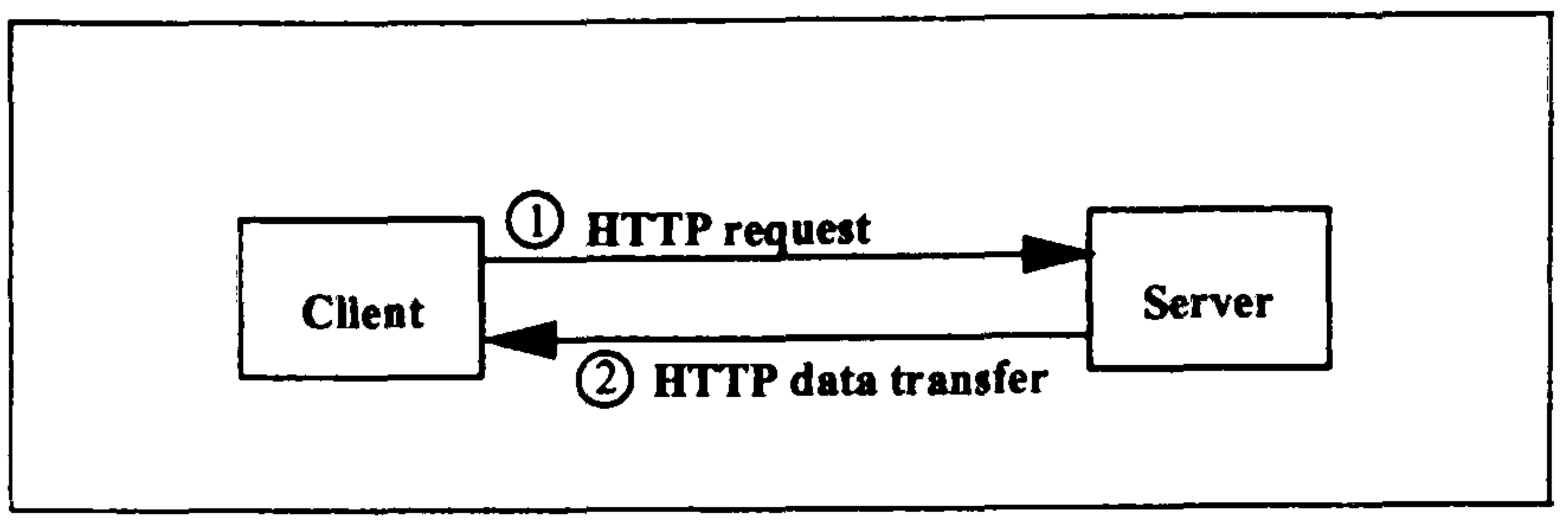


Figure 4.3: HTTP communication

This simplicity of the HTTP protocol on top of the vast Internet network is what has made the WWW successful. It currently hosts a tremendous amount of information in all sorts of fields. The advantage of the WWW lies in the ability to remotely access material from anywhere in the world, which implies the possibility of collaboration and multi-user access on a wide scale. However, the WWW is still deficient in the following three aspects:

- The spectrum of application areas.
- Standards that ensure interoperability between different applications, but accommodate competition.
- Human-computer interaction.

In order to provide applications on the web, two approaches are possible. The first approach is to publish the application files on the web and users can download these files. These applications make use of the world wide accessibility of the web. However, to be able to use such a tool one would need to have the proper software and hardware requirements of the tool. The second approach is to provide applications which are executed within the web environment. Thus, from the point of view of the user of the application, it is a portable application which requires no more than the common web tools to run. We will refer to these applications as *web-based applications*, as they utilise the full potential of the web. Thus, it is possible via the web to share applications among several users: either they can access the applications individually, or they may interact with each other in a collaboration.

Similar to the heterogeneous nature of the web-based applications, the users' computing platforms are heterogeneous. For example, a university would be accessing the web through powerful workstations with dedicated graphical capabilities at the same time that an individual would be accessing from a home PC via a modem. This situation implies two main requirements in web-based applications:

- Platform independence, which makes applications "write once, run anywhere".

- Scalable applications, which are capable of running on different hosts with different computational, graphical and peripheral capabilities without degradation in the essential parts of the application.

The advantages and requirements of web-based applications are provided by the web through its architecture and technologies. In the next section we will look at how to design and build web-based applications.

4.3 Building web-based applications

Web-based applications require communication over a network, either for the simple reason of fetching the application or for more advanced reasons like collaboration and multi-user access. Thus a web-based application consists of two layers. One layer is application-dependent and executes the specific tasks of the application. The second is a communication layer, which is responsible for translating data between the application layer and the Internet. The existence of the communication layer will depend on the architectural design of the application, while its width will depend on the communication method adopted- e.g. TCP/IP sockets or CORBA- and the complexity of required communication. Figure (4.4) shows the components of a typical web-based application.

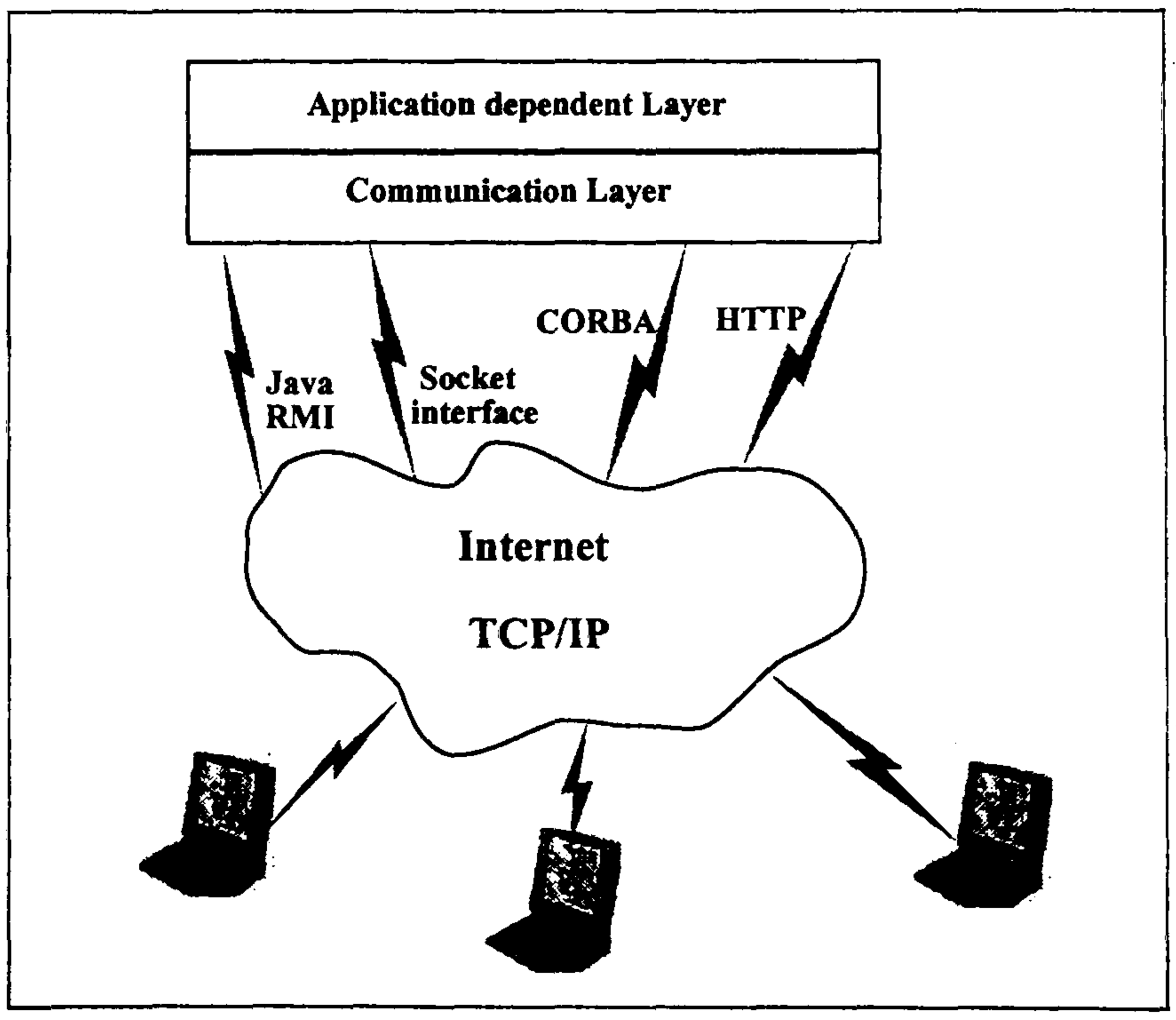


Figure 4.4: Components of a web-based application

In the next section, we will describe two options for designing a web-based application. Then we will describe technologies used to build the layers of such an application.

4.3.1 Architecture design

As described earlier, when web pages are requested by a client, a connection is made to the server via the HTTP protocol, the required data is fetched from the server, and the connection is broken until another request is issued. In this case the role of the server is to host the data and to download it whenever required. Web applications can follow a similar paradigm in what we call a *client-based approach.* In this approach, the server hosts the application and downloads it when the client requests it. All other processing of the application is then done on the client machine. This type of application is usually simple and requires low to moderate computation. A number of requirements are imposed on the application when adopting this approach. First, the application must be platform independent to ensure that different users of the web are able to run the application. Secondly, the application should be self-contained; all the data and processes required to accomplish the execution should be downloaded with the application from the server. Finally, the application should be single-user; no collaboration is supported. Under this approach the communication layer is either thin or non-existent. In the latter case the general HTTP protocol incorporated in the web browser is utilised to perform the initial download of the application.

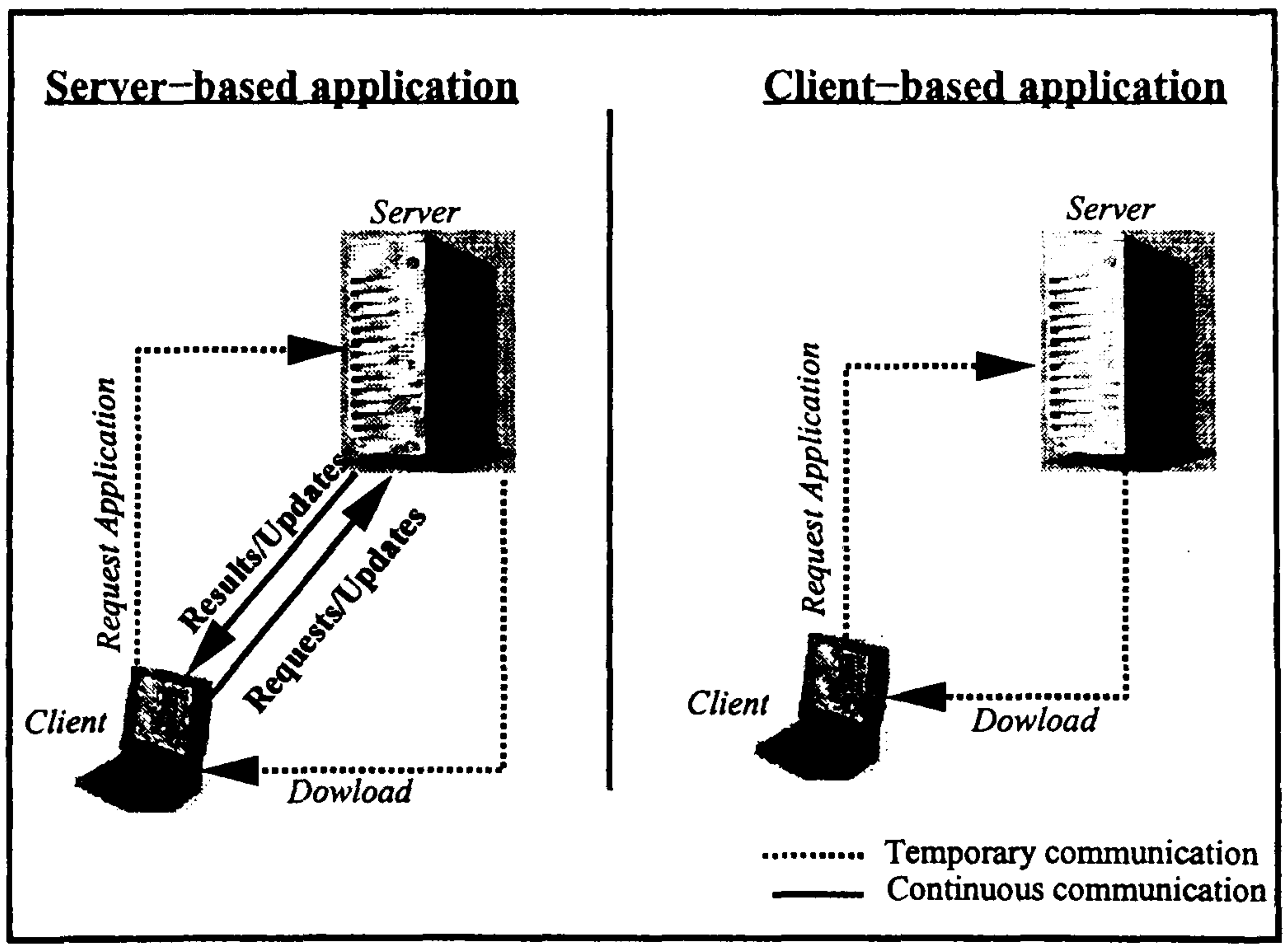


Figure 4.5: Server-based vs. client-based architecture

In order to allow collaboration, to off-load computation, or to allow platform specific processing, a *server-based approach* must be adopted. Under this approach, the connection between the server and the client is kept after downloading the application. This connection can be used to request/update data, synchronise between multiple users or perform part of the processing on the server. The server-based approach provides more flexibility; however, it can cause degradation in the application performance depending on the frequency and the size of data exchanged between the server and the client. Since these applications typically run over the Internet, network delay is expected. Figure (4.5) shows the difference between the two architectures.

In the server-based approach, both the client and the server will include a communication layer. On the client side this layer will receive data from the server and pass it up to the application layer, while sending requests to the server. On the other hand, the communication layer on the server side will receive client requests and updated data and send the requested data or updates to other participants.

4.3.2 Communication layer technologies

A number of communication protocols and paradigms have emerged in the last few years to accommodate the needs of web-based applications. Emphasis has been directed to facilitate multi-user access and collaboration over geographically disperse applications.

CORBA

CORBA is middleware developed by the Object Management Group (OMG). It facilitates a standardised framework for a distributed heterogeneous multi-vendor environment. CORBA can be used to link tools written in different languages like C, C++, Ada, Smalltalk and Java. The tools are embedded in wrappers written in the Interface Definition Language (IDL) to describe the attributes, methods and exceptions of the object. Compiling the IDL produces a stub- which is used by the client to invoke distributed objects- as well as a skeleton- which resides on the server side. The communication between the client and the server is done over the Object Request Broker (ORB)- which is the object bus. Different ORB's can communicate with each other using the Internet Inter-Object Request Broker Protocol (IIOP)[129]. The architectural design of CORBA has focused on interoperability and reusability of tools and models which already exist. A number of distributed simulation systems have been implemented utilising both CORBA and Java- e.g [94, 149, 156].

RMI

The Remote Method Invocation (RMI) provided within the Java language will handle communication between two applications written in Java acting as a client and server[8]. Applications which use RMI to communicate between processes provide a number of objects that can be invoked remotely- called remote objects. The remote object resides on the server side and consists of a stub and implementation of the methods of the class. The stub provides an interface which will define the methods that can be called by the client. When an object is downloaded, the byte code of its class is passed to the client which is executed dynamically. This dynamic nature makes RMI capable of providing generic and extensible applications.

Sockets

In addition to RMI, the Java language provides a set of classes which facilitate low-level network communication via sockets. A socket is one endpoint of a bidirectional communication between two hosts. One host acts as a server, where a socket is bound to some port on the machine. The port number- which is a 16 bit number- is used to identify a certain application running on the machine. The client side of the connection would then use the server address and the port number to request a connection to the server. When the connection is accepted, the server would create another socket which is bound to a different port to communicate with the client. Similarly, the client creates a socket that is used for the communication- see Figure (4.6). In addition, it creates an input/output stream to read/send data to the server, which is then grabbed by the server side for the same purpose. UDP or TCP protocols can be used to transmit data to and from a Java application [9].

4.3.3 Application layer technologies

Recently, there are a number of technologies that have emerged to facilitate the development of the application layer; VRML, Java and Java 3D are examples of widely used languages and file formats.

VRML

The Virtual Reality Modelling Language (VRML) is the ISO standard file format to represent 3D objects and their dynamics on the web[15]. Environments can be built using VRML by adding nodes that describe the geometry and appearance of

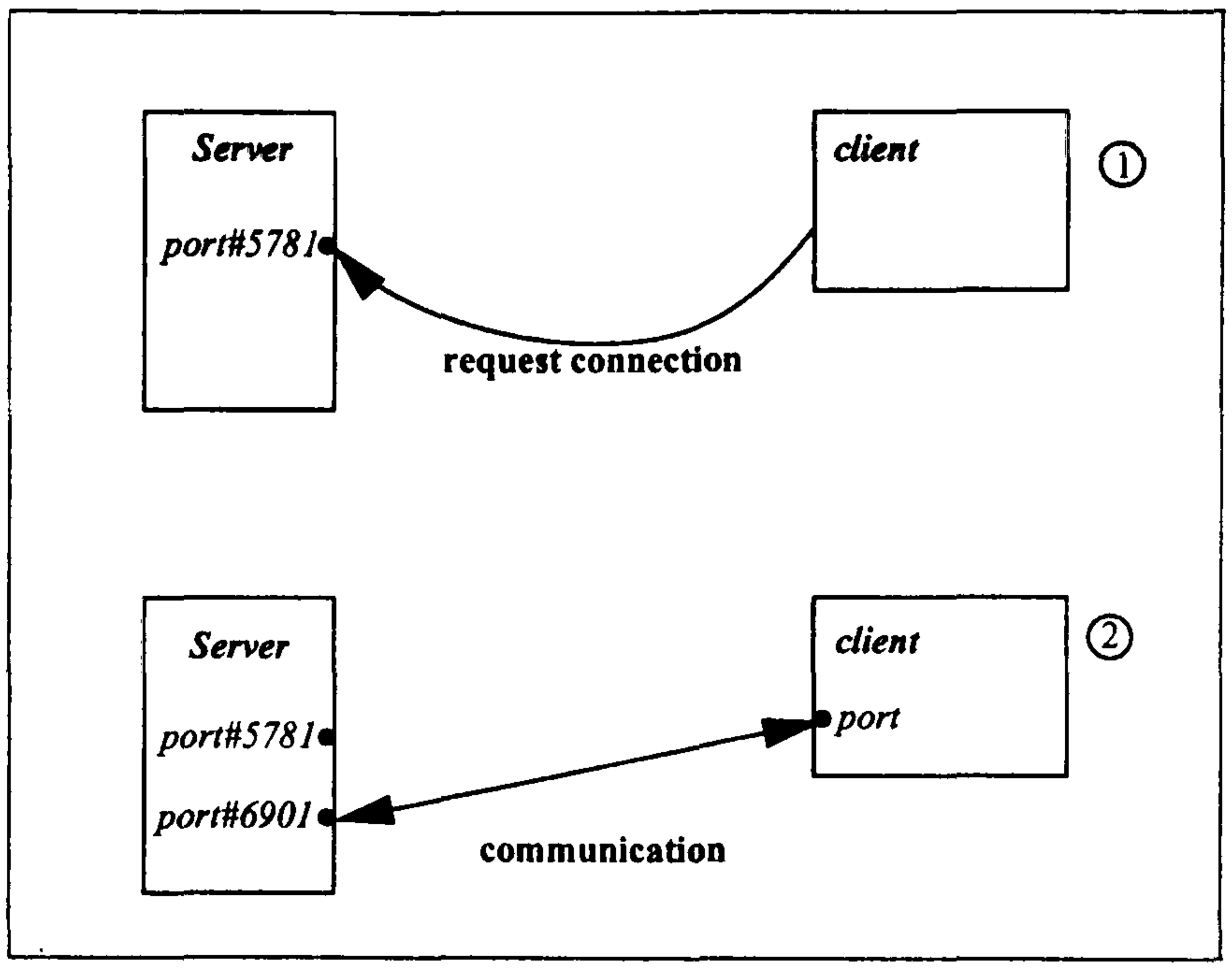


Figure 4.6: Socket communication

their objects. Each node consists of a number of fields, which describe the different attributes of the node. Most VRML nodes contain events, which are produced to indicate change in a field value. Events coming into the node are called 'EventIns', while events produced from a node are called 'EventOuts'. Nodes are grouped and nested using special grouping nodes to produce a hierarchy which is known as a scene graph. Simple dynamical behaviour of objects can be described using interpolator nodes together with sensor nodes. Interpolator nodes provide keyframe animators, while sensor nodes are sources of events which can detect user inputs. Events are routed from one node to another in order to create reactions to user interaction. In addition to the nodes provided by VRML, the user has the ability to customise these nodes to create new ones using prototypes. When creating a prototype the fields and events of the new node are specified, then the nodes which comprise the new node are specified. Multimedia files such as audio and video, and hyperlinks to other web resources can be incorporated in the VRML environment.

Although VRML is useful in exploring data and models in three dimensions, the amount of interaction with the model that it provides is not sufficient to cover all user needs. Efforts have been made to extend the VRML capabilities by adding tools which support multi-user access, dynamics and collision [41, 39, 92]. In these examples, to facilitate the communication between the tool and the VRML world, they have developed their own browser. This method of extending VRML is not ideal, since it is non-standard. Another way of extending VRML is through the use of Java to provide a software component to the VRML file. There are two ways to do this: either Java can be incorporated within VRML in script nodes, or one can use the External Authoring Interface (EAI).

A script node receives events from the virtual scene because of user actions or change of state of some nodes. The script code which can be written in JavaScript or Java processes these events and outputs new events that change the status of other nodes.

The EAI is an extension of the VRML language to provide access to VRML worlds from other environments. Initially Java was the only language supported by the EAI. The EAI can connect the VRML world to a Java applet which is embedded in the same HTML page and runs alongside. Using this interface the Java applet can send 'eventIns' to the VRML world affecting the state of its objects and receive 'eventOuts' of the latest updates in the world. In the latest release of EAI (Jan 1999) both Java and Interface Definition Language (IDL) are supported [3].

Java

Java [9] is a high level object-oriented language that fulfills the platform independence requirement of the web. Java can be used to create simple web-applications using Java applets, which are Java code that can be embedded in an HTML page. Java is an interpreted language, where a compiler produces a byte code which is simple and machine independent. The byte code is then translated into machine code during the run time. This feature together with the portable specification of the Java structures is what gives Java its strongest characteristic- being platform independent. On the other hand, because it is an interpreted language its performance does not match other compiled languages-such as C and C++. This can be overcome by utilising optimisation options during compilation which produces machine code. In addition, Java is designed to work over networks in distributed applications, hence security issues have been tackled in Java. Indeed, it is a powerful language for creation of stand-alone applications. However, the amount of graphical support in Java is minimal. This is what makes the cooperation of Java and VRML a strong tool in the building of web applications. Recently, Java 3D has emerged as an extension to Java to support the construction and manipulation of 3D graphical applications [6].

Java 3D

Java 3D API's are used to build a scene graph which contains a description of all the objects contained in the environment and their behaviour. Behaviour can be incorporated by extending the behaviour class with Java code. Java 3D provide three types of rendering with different levels of optimisation to provide efficient performance across different platforms. In addition, it supports the import of different file formats like VRML via special loaders- that translate the objects in the file to 3D Java objects. Furthermore, Java 3D gives support to immersive environments by providing a 3D sound class and input device classes which can be extended to cover any peripheral [6]. In fact a demonstration has been developed jointly by Fakespace Inc. and Sun to utilise Java 3D in building a 3D virtual environment for furniture layout. The virtual environment uses the immersive workbench and data gloves [4].

All these technologies have been utilised in the last two years to create web-based applications. In the next section we will review some of the applications that have been offered within the web society and we shall attempt to find a suitable classification for them.

4.4 Web-based applications: Review

There are already a surprising number and variety of web-based applications. Although the field is very new, it is growing exponentially. Therefore it is useful to attempt to define categories to classify these applications. We can classify web-based applications according to their field of application or to the architecture adopted- i.e. server-based vs. client-based. But here we suggest another classification which may give more insight.

We propose a classification for current web-based applications based on the goal of the web society. As previously mentioned, the ultimate benefit of the web is to satisfy the individual needs via a vast computer and communication architecture. Thus, we were able to identify communication, application and HCI as contributors to this society. Hence, we will classify the web applications according to the degree of satisfaction of these three areas. Thus, the communication aspect refers to the degree of collaboration and sharing the web application allows. Meanwhile, the application aspect is a measure of the level of sophistication of the web application. Finally, the HCI aspect refers to the degree to which the simulated interaction mimics the real interaction. All these aspects should be judged within the requirements of a certain field.

Figure (4.7) represents this classification as applied to the simulation field in which we are interested. In simulation, the application is modelling a system where we are interested in its behaviour. Hence, modelling the geometry and the behaviour

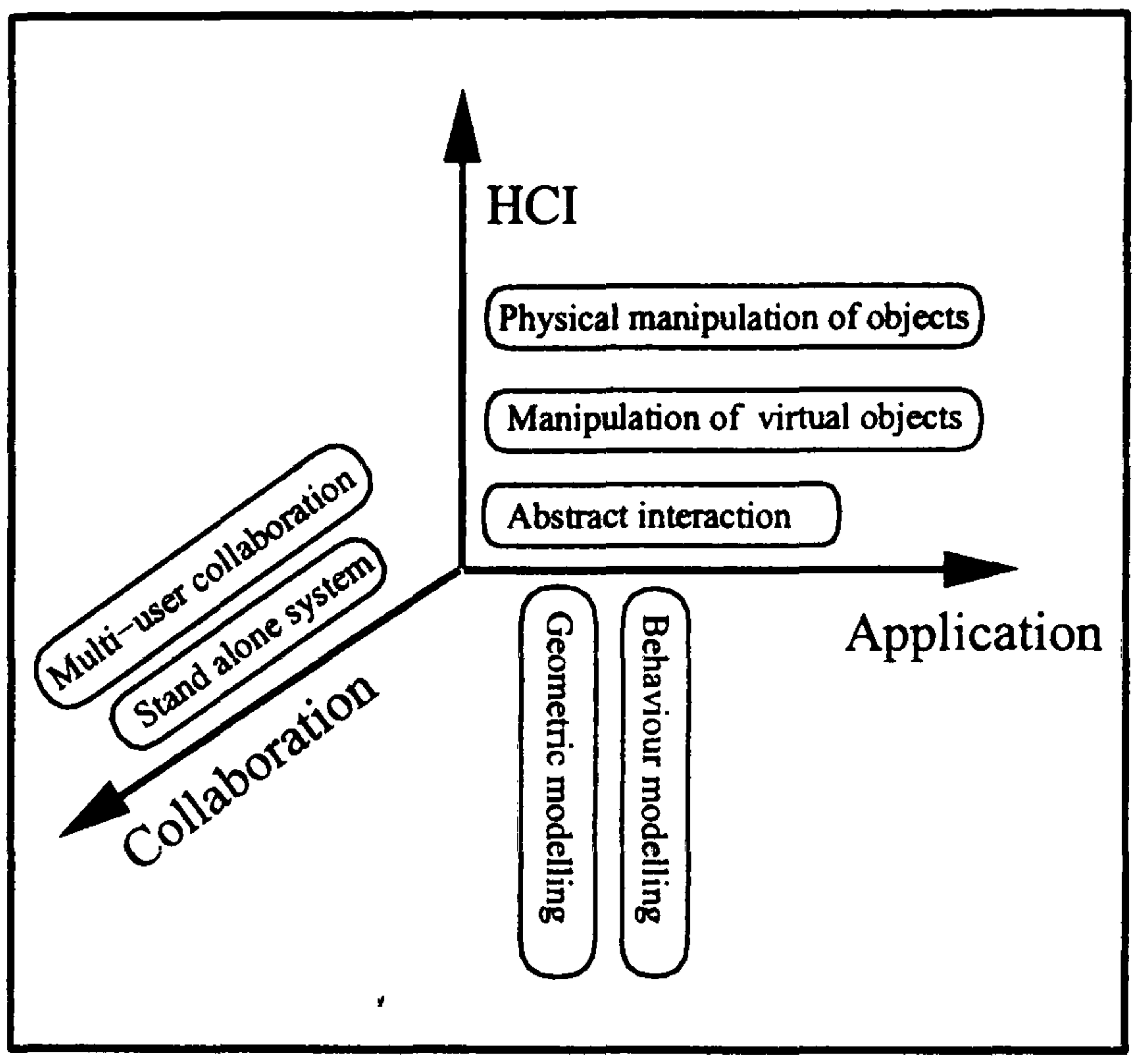


Figure 4.7: The classification criteria of a web-based application

of this system are two important aspects of this type of application: geometry is essential in a visual simulation, while the degree to which behaviour is modelled will distinguish different approaches. These two aspects create the "application axis". The HCI axis allows us to distinguish three levels of interaction: at the lowest level is what we term abstract interaction, where there is no attempt to simulate the physical interaction of the real world, rather this is replaced by the simple control of some abstract widget (such as a slider); a higher level is interaction via some direct simulation of a real physical device (such as a complex surgical instrument) - this we term manipulation of virtual objects; and finally at the highest level, the real physical devices are connected to the simulation and used to interact with it. The collaboration axis distinguishes between simulations that are restricted to a one-user world, and those that allow multi-user participation. Of course this is a broad generalisation, but the diagram of Figure (4.7) gives us a means of illustrating the sophistication of a simulation by positioning it in the space defined by these axes: the further along an axis, the more elaborate is the attempt to simulate reality. In the rest of this section we will review some of the existing web-based applications. We will attempt to position them relative to the HCI and the collaboration axes. Due to the variety of fields of these applications, we will not attempt to classify them according to the "application axis". Figure (4.8) shows the categories of these applications in the two dimensional HCI/collaboration space.

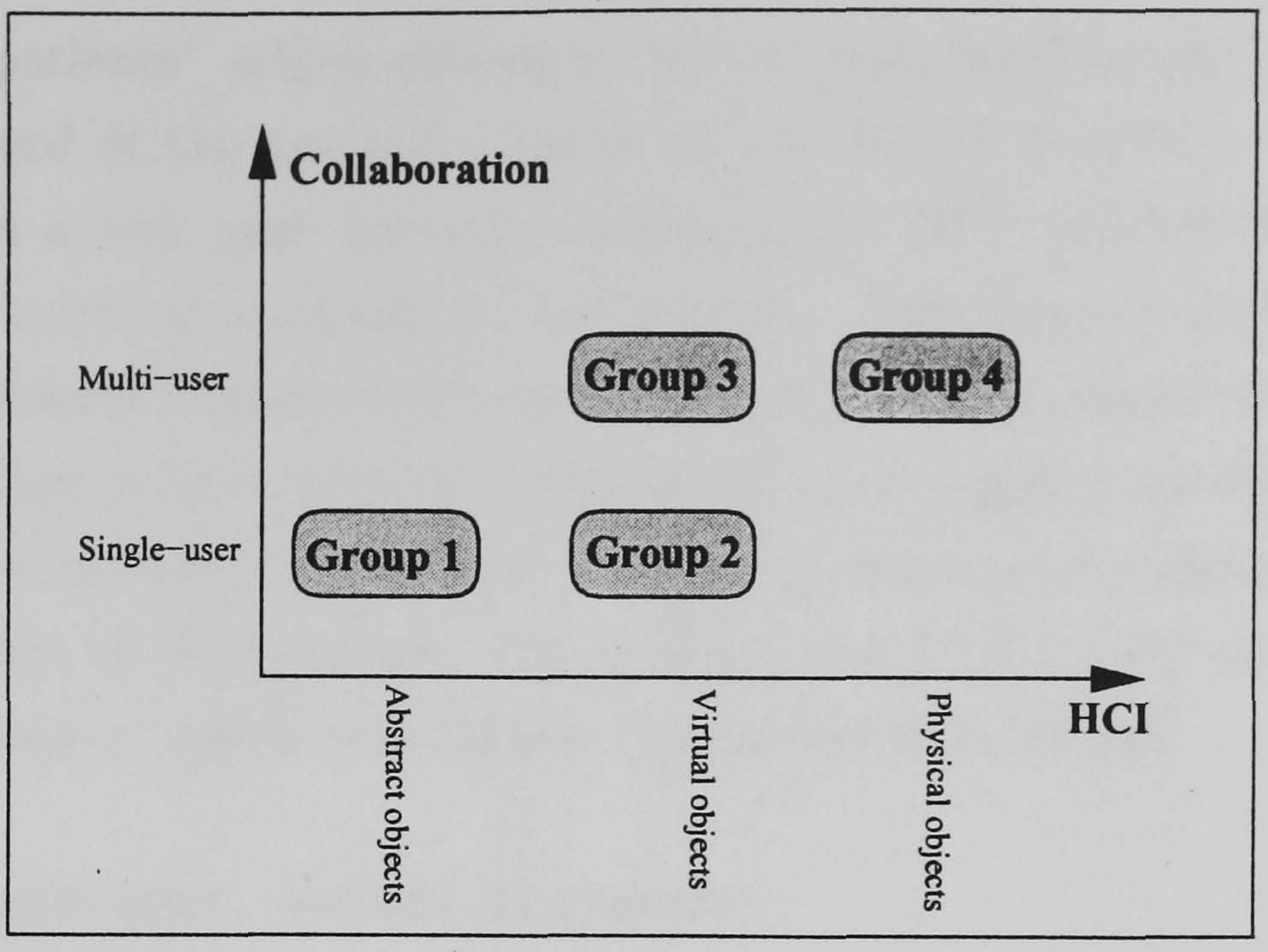


Figure 4.8: Two dimensional categories of the web-based applications

Group 1 - Single-user, abstract interaction

Group 1 are single-user applications that are characterised by abstract interactions such as text forms for input and dynamically updated images as output. The early examples of web-based applications are characterised by this type of interaction. Utilising the Common Gateway Interface (CGI) and a script language such as Perl, these applications communicate with the server over HTTP to query a data base- e.g. [178]- or to process user input, such as the student answers in a web-based educational package- called CyberProf- which utilises a sophisticated grading software on the server side [91]. More recently, the same paradigm has been used in the "web-based physical modelling for multi-disciplinary engineering" which uses the DYNAST simulation package to solve an engineering problem formulated by the user[117]. A form is used to input the problem and the output is presented in a graphical form using Javascript, or by email. The type of problem solved by this system requires substantial processing, which calls for a server-based solution. Another example is the "web-based robot arm control" system which accepts the new position of the robot arm from the user in a text format or a mouse click on an image. A real robot arm is moved to the new position and an image of the modification is captured and sent back to the user[64]. Again a server-based solution is employed in order to handle the inverse kinematics calculation.

Under the umbrella of the Electronic Medical Record System (EMRS)- mentioned in section (2.2)- a number of applications make use of the web to facilitate medical services for physicians. An example is the "web-based monitoring system for intensive care unit patients" which utilises a "server-push mechanism" to dynamically update the record of the patient's status on the web in near real time[182]. The system provides a web page for each patient in the ICU which reflects the charts of the bedside monitor attached to the patient. This project makes the current status of the patients together with their medical records available for the physician remotely. Another project provides an immunisation registry system for clinicians which allows the access of a database containing information about the timetable of the vaccination of the patient. This system uses CGI scripts to allow different health institutions to access and update the patient records [98].

Group 2 - Single-user, virtual interaction

This group has a more advanced form of interaction which involves direct manipulation of objects in a virtual world, while reflecting output as a change of the objects' attributes. A common factor among these applications is the use of VRML to represent the 3D objects of the application. For instance, the "3D visible human"[12] system allows the visualization of medical images by sweeping a selector across a 3D model of the human body. In addition, the "web-based ventricular catheterisation" [96] allows neurosurgeons to manipulate a cannula to rehearse a ventricular catheterisation procedure - Figure (4.9) shows a snap shot of this application. In a web-based medical visualization system provided by the surgical planning laboratory (SPL), Java 3D is used to develop an interactive system which allows the physician to interact with anatomical models. In addition to the 3D models, the user is provided with some manipulation tools- to translate, rotate and zoom the model- slice viewer- to view data in 2D- and other tools to control the colour and transparency of the models [21]. Another web-based surgical planning system is developed by Stanford University Cardiovascular Biomechanics Research Laboratory [17]. The system consists of a Java applet and a VRML world. The Java applet provides the user with a 2D geometry of the aortic vessel and diagrams of the pressure and speed of blood at different positions of the vessel. It also provides the user with a set of surgical options to choose from. Simple 2D graphics is used to simulate the chosen treatment. On the other hand the VRML world provides 3D geometry of the aorta in addition to the capability of visualizing the pressure and speed of blood flow on this geometry. Since the calculation of blood flow is computationally expensive, all these calculations were pre-calculated and made available with in the system. These applications are examples of client-based systems.

Figure 4.9: Web-based ventricular catheterisation system (Extracted from: http:// neuronott.nottingham.ac.uk/ WEBpages/VirtualSurgery/)

Under the same category some applications choose the server-based design to fulfill special requirements of their applications. The "web-based segmentation and display of medical images" [158] uses a Java applet to accept segmentation parameters and output a 3D polygonal model of the segmented data. The segmentation computation is complex, and executes on a server. The "air quality web-based data visualization" provides access to data which show air pollutants at certain sites in Britain. The user choices are accepted through a form, and passed via a CGI script to the IRIS Explorer visualization system. Output is produced as a VRML world consisting of a surface or a graph representation of the data [188].

Group 3 - Multi-user, virtual interaction

Similarly, applications such as "JackMoo" [160]- a 3D human simulation for training purposes- and "web-based combat simulation"[38]- distributed simulation of military exercises- follow a similar paradigm of directly manipulating virtual objects for interactions. However, they support multiple user participation in their environment, which makes them more advanced than the previous examples along the collaboration axis. In the JackMoo project- carried out at the University of Pennsylvania- the client side contains a Java applet which is responsible for communicating text commands to the server via sockets. On the server side a lambdaMOO program is responsible for multi-casting actions to the participanting clients. The lambdaMOO is a networked, multi-user programmable system which consists of a database- containing objects of the environment- and a server which handles client messages. On the other hand, in the group distance exercise system, RMI is used to communicate between the server and the clients- called equipment simulators. The server is responsible for multi-casting actions among the clients as well as keeping an up-to-date representation of the world and the participants. Clients are of two types, either intelligent units- which are equipped with radar screens and can detect fighters- or combat units- which receive information from the intelligent unit and launch missiles.

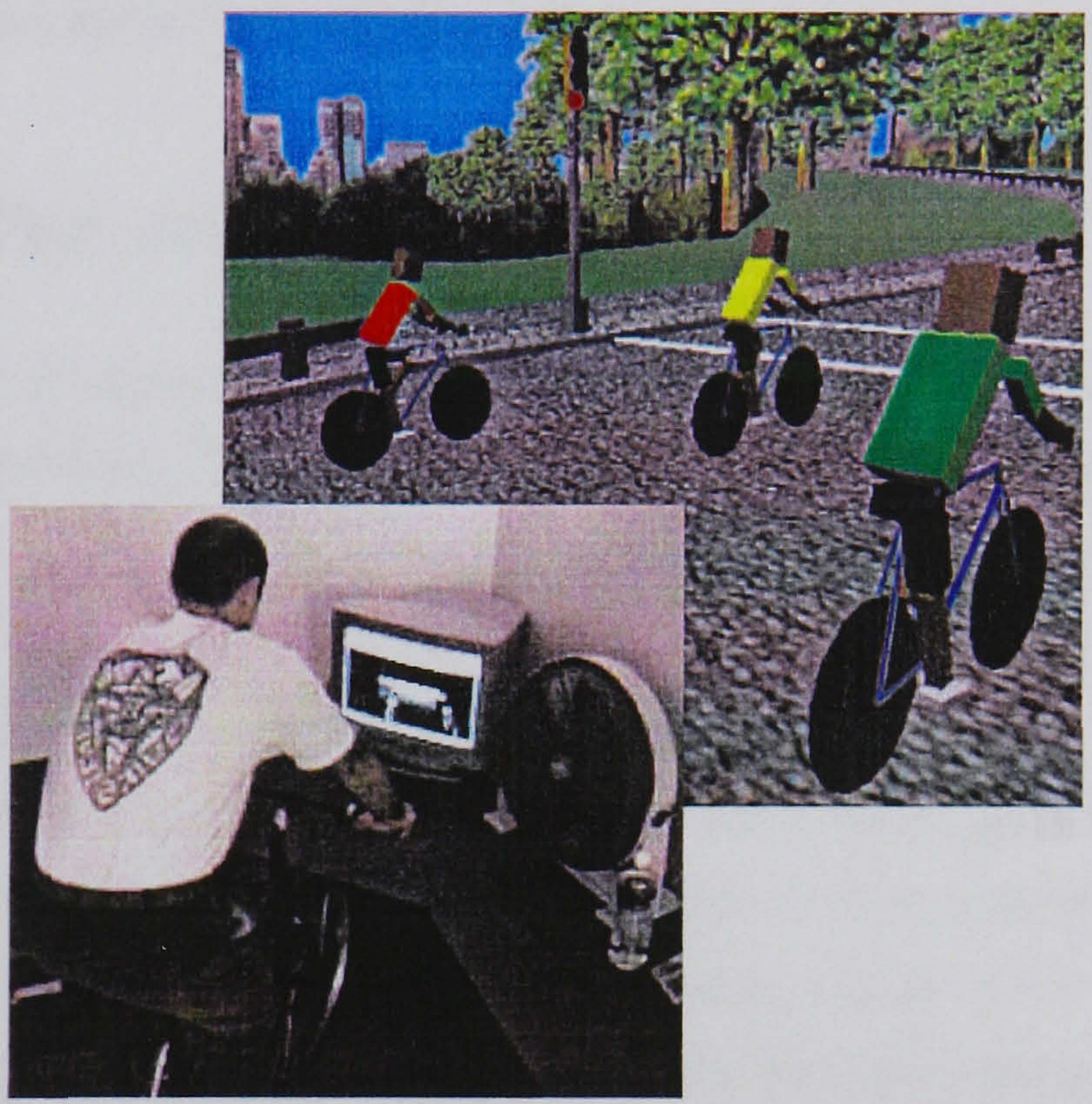


Figure 4.10: The Peloton bicycle riding (Extracted from: http://www.multimedia.bell-labs.com/ projects/peloton/)

Group 4 - Multi-user, physical interaction

At the highest end of the HCI spectrum we find web-based applications that accept direct physical interactions with the user. They use a more natural interface of gestures and direct manipulation of real objects for input, while visual, audio and tactile feedback are used for output feedback. An example of such an application is the "Peloton" project [44], which provides a virtual environment for bicycle riding on the web - Figure (4.10). In this project a sensored bicycle is used as an input device. The user receives visual and force feedback as well as wind blowing - via a fan - to give a sense of the speed. In addition, the application supports multi-user participation in the environment. In this application in order to provide access to unconventional input/output peripherals, the developers explored several options. This is due to the fact that a client side Java applet is not permitted to communicate with any other hosts except the originating host- due to Java security design features. This led them to send peripheral input to the server side, then the client would receive the input from the server. This solution is inefficient and produces heavy network traffic. Another solution is to put the client class on the local class path of the client host. In addition, in a later release of Java (JDK 1.1), signed applets have been granted the ability to connect to a host which can be described as safe. Both these solutions have their disadvantages. A future plan of the Peloton team is to explore CORBA as an option since it facilitates heterogeneous network communication.

4.5 WebSTer requirements

In this thesis we are studying the feasibility of providing a surgical training application on the web (WebSTer). WebSTer (Web-based Surgical Trainer) is an application to train radiology trainees on interventional radiology procedures. In this section we will identify the requirements of this application from the web prespective.

From the training point of view, this application targets to transfer the skills listed in page (40) to trainees. Meanwhile, from the system point of view, it targets the two requirements listed in page (41): realism and accessibility. According to our training requirement list, we can see that some skills- such as procedural and automised skills- require the early instructions and demonstration stages of training to introduce them to the trainee. Meanwhile, skills such as the representational and decision making require practice to acquire. Hence, our solution should support two main stages of training: the *background stage*- which includes instructions and demonstration- is the stage where the trainee acts as a recipient of information; while the *practical stage* involves the trainee's interaction with a controlled training environment. Figure (4.11) shows the two main parts of our training processs.

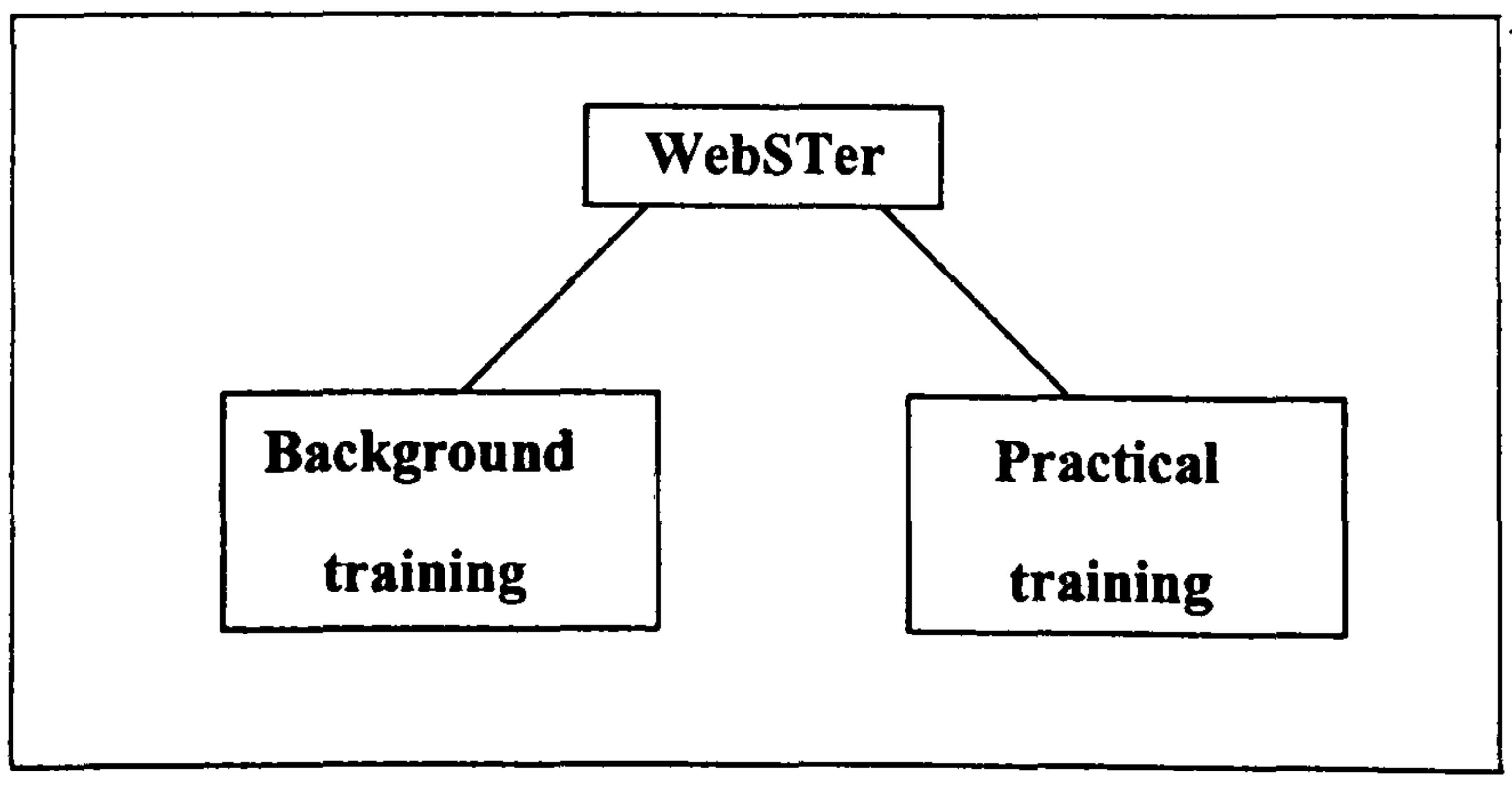


Figure 4.11: Two stages of training supported by WebSter.

In order to support the background stage of training, the usual 2D web technologies- e.g. hypertext and movies- are sufficient. Thus, this part of training is using the web as an information repository. However, the practice environment will require the utilisation of 3D technologies to provide a realistic training simulation. Our review of medical applications- in section (2.2)- has shown that three dimensional representation of the anatomical structures provides better understanding of the data. In addition, we found that providing physical and physiological modelling of the anatomy and the surgical tools together with intuitive interactions assist in building the realism factor of the environment and consequently enhance the transfer of skills. Hence, we would like to investigate the degree of current web support for such training environments which require substantial computation for modelling purposes and complex and special interactions.

Based on our earlier definition of a web-based application, we found that such an application needs to be platform independent and scalable to accommodate the heterogeneous nature of the web environment. This implies that both our modelling and interactions should be complying with these basic web requirements. Again looking back at the medical review, we find that the complex interaction was provided for via special peripheral devices- which are hardware-dependent. Meanwhile, the high computational cost of modelling was accomplished by dedicated hardware and special software. Furthermore, from our review of web-based applications- section (4.4)- we saw applications which require high computation - such as "segmentation of medical images"- have met this requirement by adopting a server-based approach. Meanwhile, "Peloton" provided a server-based approach as one solution to the special peripheral requirement of their application. Thus, the requirements of our application call for a server-based architecture, where platform dependent components of the application are placed on the server side; while the client communicates with the server whenever data or processing is needed from these components- see figure (4.12).

Adopting such an architecture will immediately introduce network communication as a key factor in the application. In order to characterise this communication, we need to examine a typical scenario of such application. Suppose that the physical and physiological modelling of the blood vessel and the surgical tools are carried out on a dedicated high capacity server, while the virtual environment interface is downloaded to the client side. When the user interacts with the virtual environment by inserting the surgical tool into the vessel and applying forces to it, the client side will ask the server for instructions about how to behave based on the given user action. On

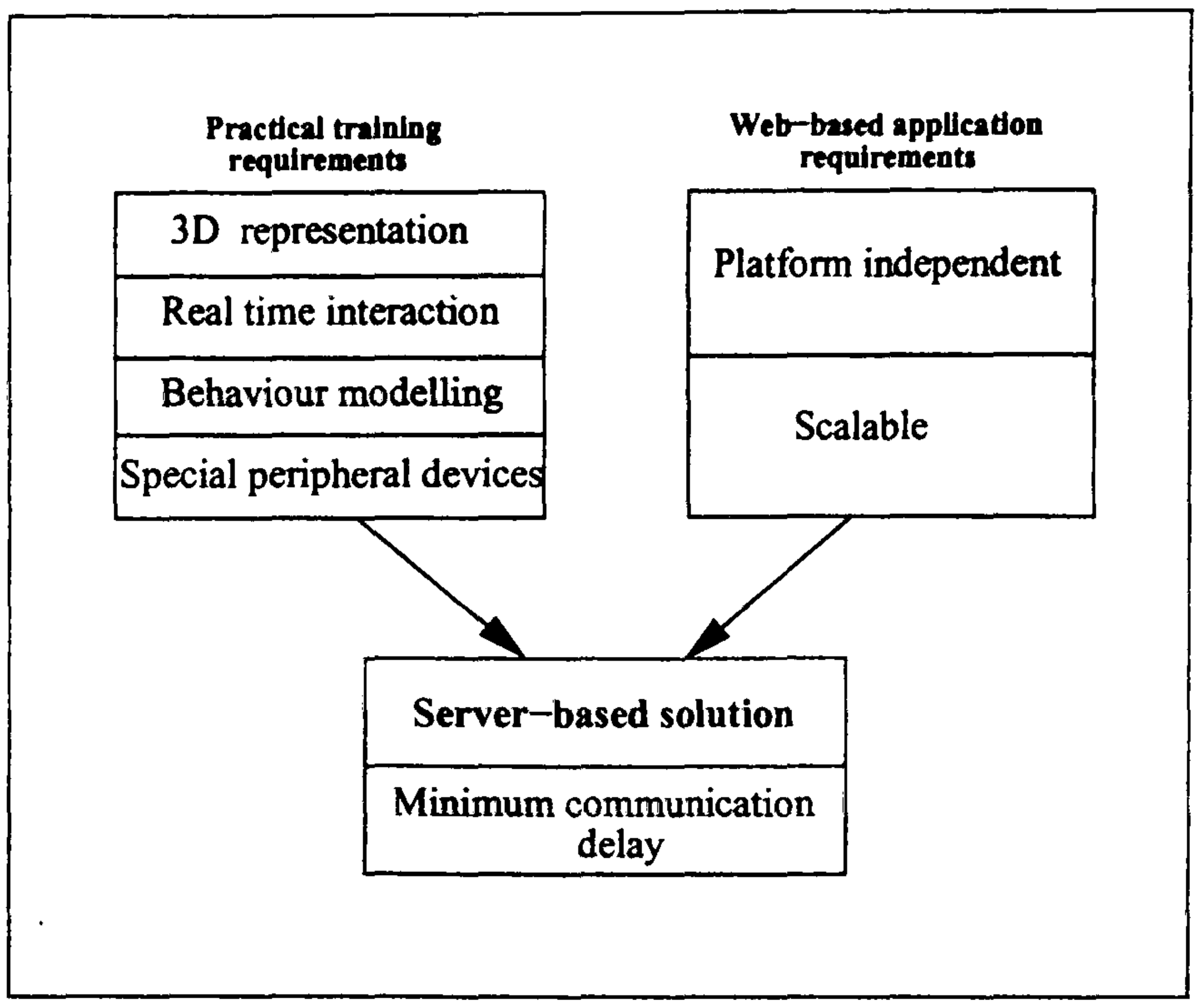


Figure 4.12: Web requirements of our training system

the server side, the modelling component performs complex computations based on the data sent and provides the client with the answer. This scenario should take place within milliseconds to provide the user with an interactive response. Thus, we can classify this application based on the degree of tolerance of delay- which is reported earlier in section (4.2)- as isochronous. Isochronous applications are described as "Time sensitive to an extent that delay adversely affects usability". Furthermore, the amount and shape of data exchanged between the server and client will depend on the user interaction; thus this application transmits bidirectional burst data. Hence, in addition to the general requirements of web applications-platform independence and scalablility- our surgical training application will require minimum communication delay. Thus, it would benefit from a quality of service (QoS) communication. Figure (4.12) summaries the requirements of our web-based application based on the training requirements.

4.6 WebSTer environment

4.6.1 Background training

In order to support the early stages of training in our application, we have created some web pages to give instructions and demonstration information. The instructions web page includes description of the anatomy, surgical tools used and the steps of the procedure. Another page provides a set of movies, which were captured when we practised the manipulation of a guide wire and a catheter in a silicon mockup of the aorta [18]. The goal of these videos is to demonstrate the procedure of manipulating the surgical tools as well as to give an idea of the difficulties expected. In addition, we have provided a set of links to outside information on the web that might be of interest to the trainee. The web pages provide a description of our training simulator to orient the trainee in the use of its interface, hence creating a link to the practical training. A number of CT scan data sets have been obtained from St. James's University Hospital of patients suffering from AAA. The data sets have been prepared- details of data preparation is in chapter (6)- and the different models are made available for the trainee to choose from.

In order to support the decision making skill, we provide a procedure evaluation form. The trainee is asked to perform the measurements required to decide the suitablity of the case for an endovascular procedure- i.e. aneurysm neck, distal neck and aneurysm length- using our practice environment. Based on these measurements the trainee makes the decision of performing the procedure and adds comments of any expected difficulties and the type of surgical tools used during the procedure. The contents of the form can then be kept in a file to evaluate the trainee.

Furthermore, one way of developing the representational skill of the trainee is to display the CT slices consecutively in the same way radiologists do in real life. The trainee would scan through these slices and attempt to build a 3D mental model from them. A link to the 3D geometric model- in VRML format- is then given at the end to verify the mental model.

These web pages have been developed to give an example of how the web can offer multimedia to support the various training stages- Appendix (C) contains these web pages.

4.6.2 Practical training

In the requirements section we deduced the need for a server-based architecture in our solution. Furthermore, we recognised two main parts of the system: a user interface and a simulation engine. The user interface will facilitate interactions with the environment, while the simulation engine will compute the environment reaction to the user manipulation as well as any internal actions that the model might exhibit. In WebSTer, the simulation engine is a physically-based model, which simulates the behaviour of the surgical tool- chapter (6) will describe this component of the system. Meanwhile, the user interface will be described in chapter (5). Under the server-based architecture, we have placed the physically-based model on the server to ensure efficient execution, while the user interface is downloaded to the client side.

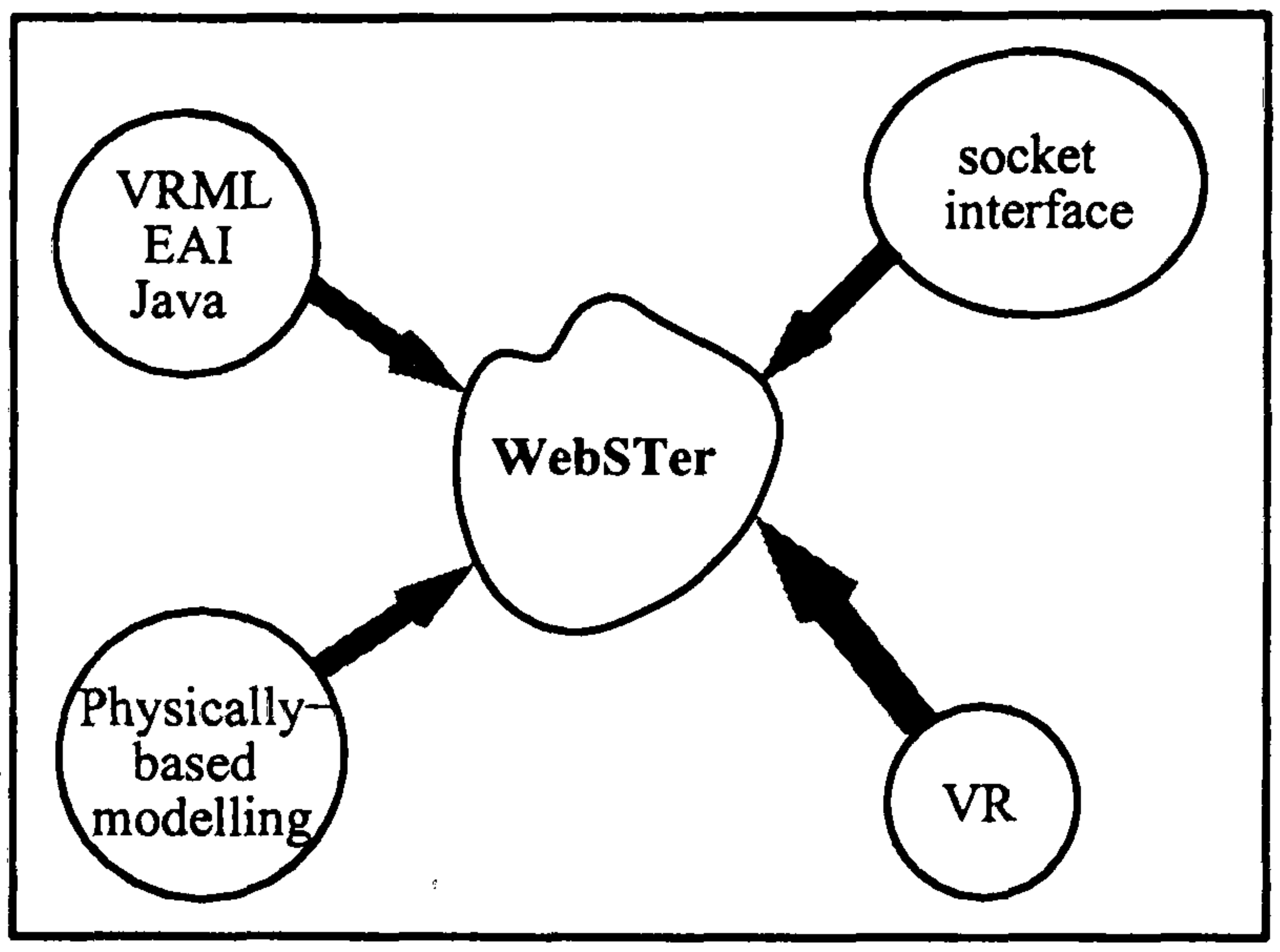


Figure 4.13: Web technologies utilised in WebSTer.

Looking back at the technologies involved in building a web-based application- Figure (4.1)- and adapting this figure according to our own implementation- Figure (4.13)- we find that physical-based modelling is the basic technology we are migrating to the web environment, while virtual reality interaction is the HCI technology that we provide for our application. Meanwhile the communication technology that we utilise is the socket interface from the Java language. This is due to the fact that both the server and the client side in our application are Java-based. Thus, we do not need a complicated communication paradigm such as CORBA, meanwhile RMI was not available at the time work started in this project. Transport Control Protocol (TCP) is used to send messages between the server and the client. The TCP was used because most messages exchanged between the server and the client- except for time messages- require guaranteed connection. For example, if the user applies a push force to the surgical tool, this event will affect the result of the simulation. If a UDP is used instead and the message is lost, all the subsequent reactions are invalid. A Java applet on the client side is responsible for establishing the socket connection with the server- as described in page (50). Meanwhile on the server side, a Java application is responsible for receiving the client messages and sending them to the simulation engine. Figure (4.14) is an adaptation of Figure (4.4) and shows the communication between the two 'communication layers' of our client and the server via sockets.

Furthermore, we utilise the 'application layer' technologies to provide for a platform independent and scalable user interface- since it will be downloaded to the

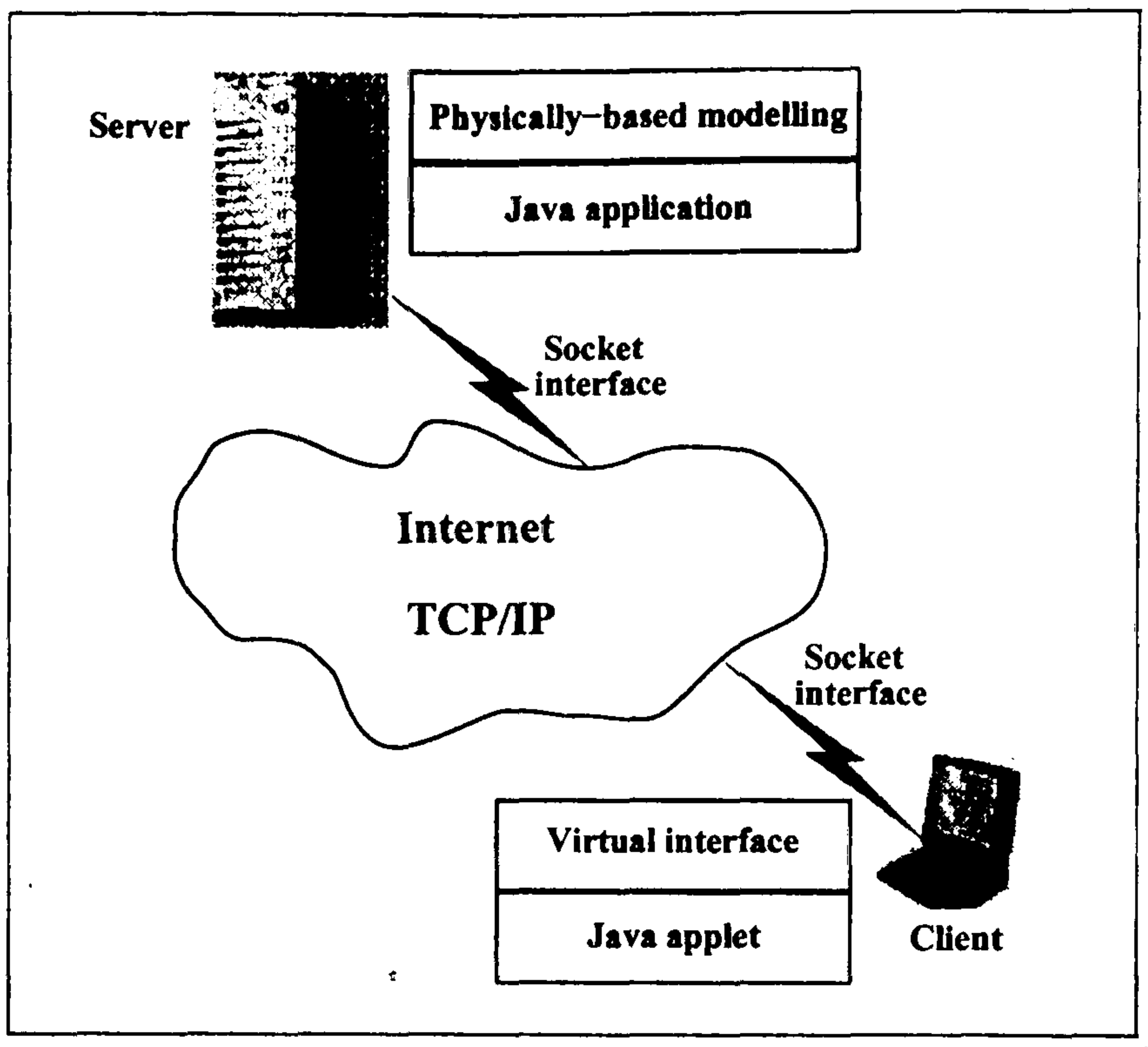


Figure 4.14: Communication between WebSTer client and server over the socket interface.

client side. The user interface utilises VRML to provide the 3D rendering and the manipulation needed for the virtual training environment. Simple dynamic actions in this environment were provided via the scripting capabilities of VRML. Meanwile, Java applets were used to provide parts of the functionality that VRML failed to support. Some interactions which required synchronisation between different VRML worlds and Java applets were provided through the EAI API's. In addition, the EAI has been utilised to update the VRML worlds with results that come from the simulation engine on the server, while user manipulations are routed through the EAI to the Java applet responsible for communication with the server- see appendix (A) for details of this implementation. Figure (4.15) shows the architecture and the communication between the different parts of the training system.

Since Java has been used to implement most parts of our surgical training system, classes have been used to represent objects in the environment as well as their behaviour. The two layers of the web-based application- communication and application layer- apply to our training system. Two classes- a client and a server class- form the communication layer. Together both classes are responsible for opening a socket connection, creating input/output streams and creating listening threads. The application layer consists of two layers: the *virtual clock layer* and the *simulation layer*. The virtual clock layer consists of two classes, one on the server and the other on the client side. Initially, the client sends the value of its clock to the server to note the difference between the clocks of the two systems. A Java thread is then

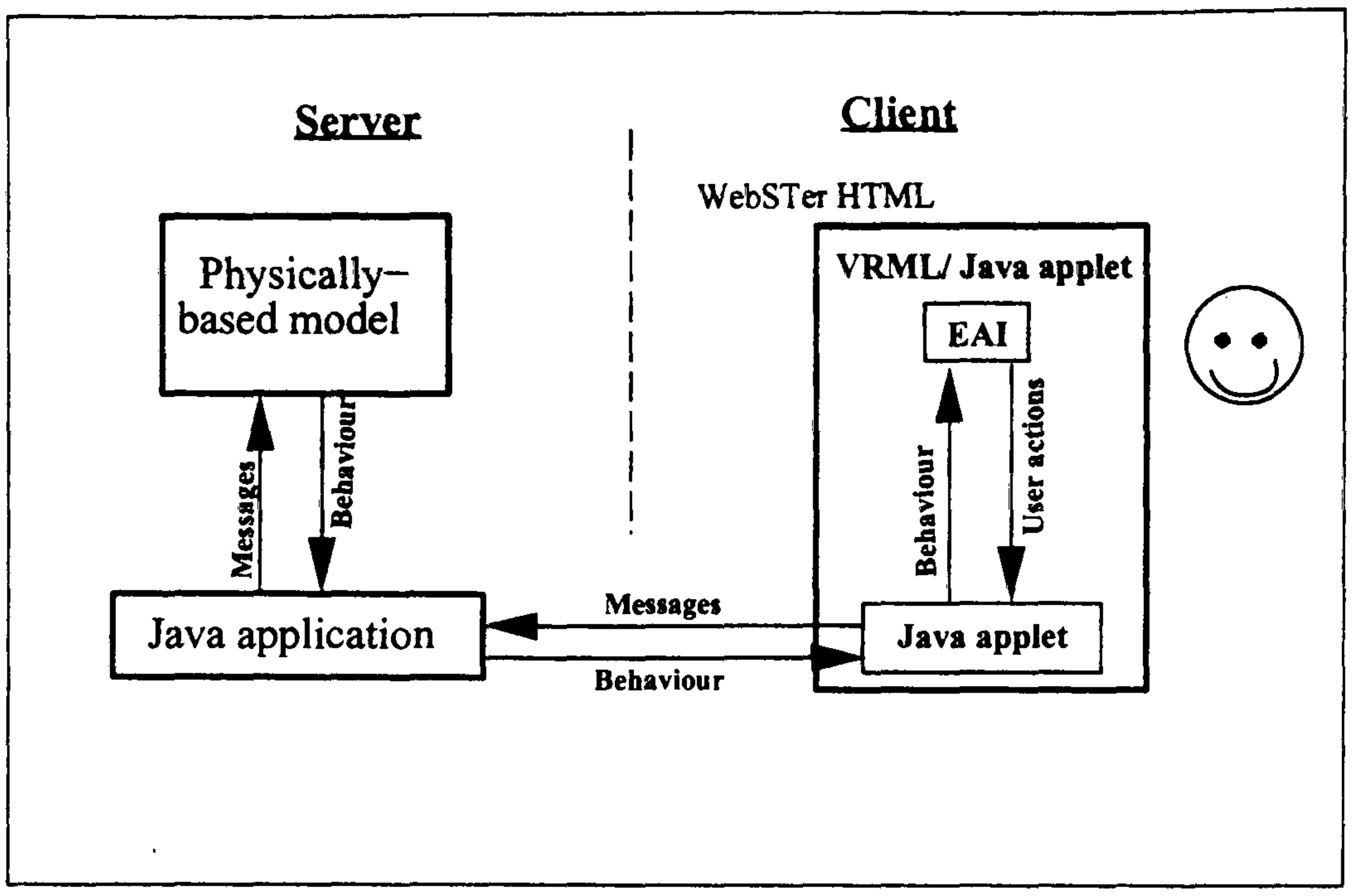


Figure 4.15: The architectural design of WebSTer

created on the client side to produce time events every interval- which is specified in the application- and to send time stamps to the server side. Upon receiving the time command on the server side, the server will compare the client time to its own clock taking into account the difference between the two clocks. If the difference between the client time and the server time is greater than the interval size, the server will disregard this message, otherwise it will pass it forward to the simulation engine. This mechanism has been chosen to relieve the simulation engine from calculating expired time messages, which would otherwise delay on-time messages.

To synchronise between the client and the server clock, we utilise the "Date" class provided by Java. The advantage of using this class is that it provides Coordinated Universal time (UTC). Thus, the server and the client could reside anywhere on the world while their time commands are unified. Meanwhile, the simulator layer consists of a set of classes which represent the objects of the training environment- i.e. the surgical tool and the blood vessel. In addition, other classes are used to implement dynamic rules needed to simulate the motion of the surgical tool; details of the simulation layer classes will be presented in section (6.7.4).

Communication between the layers of the application is message-based. Messages are exchanged at the lowest level- i.e. communication layer- between the server and the client. Messages that are not interpreted by one layer are passed forward to the next layer. Figure (4.16), shows the three layers of our system and the functionalities of the first two- shown under the horizontal arrows. It also shows the messages passed- both on the client and the server side- from the bottom layer to the top. Thus, the communication layer will pass all messages to the virtual-clock

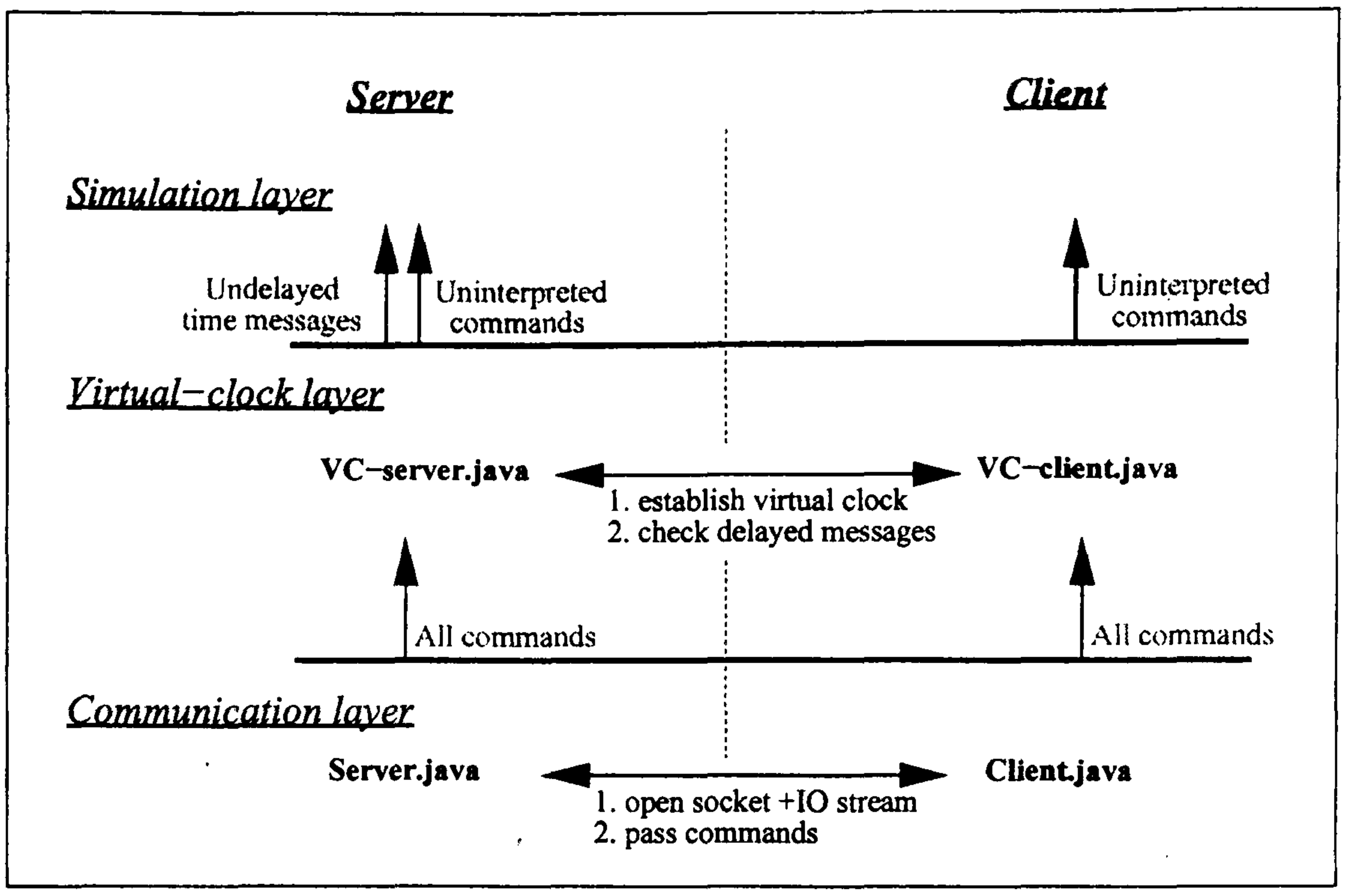


Figure 4.16: WebSTer system layers

layer, while the latter will pass undelayed time messages and commands that are not recognised by this layer to the simulation layer.

4.7 Discussion

In our surgical training solution, we were able to support a computer-intensive application by adopting a server-based approach, which ensures efficient implementation without sacrificing platform independence. In addition, the surgical training required specialised input/output devices to accommodate the complex interactions. We were not able to deliver more than the usual interaction peripherals- mouse and keyboard- for our application. This is due to the fact that no browser supports any other devices. Work is going on by Hand to provide web support for other interaction devices- e.g. HMD and 6 degree of freedom input devices, however the resulting solution is based on an in-house browser rather than a standard one [63]. There are other ways around this problem- as we saw in the Peloton system- however they are not efficient yet. With the recent availability of Java 3D, it will be possible to support other devices in the near future.

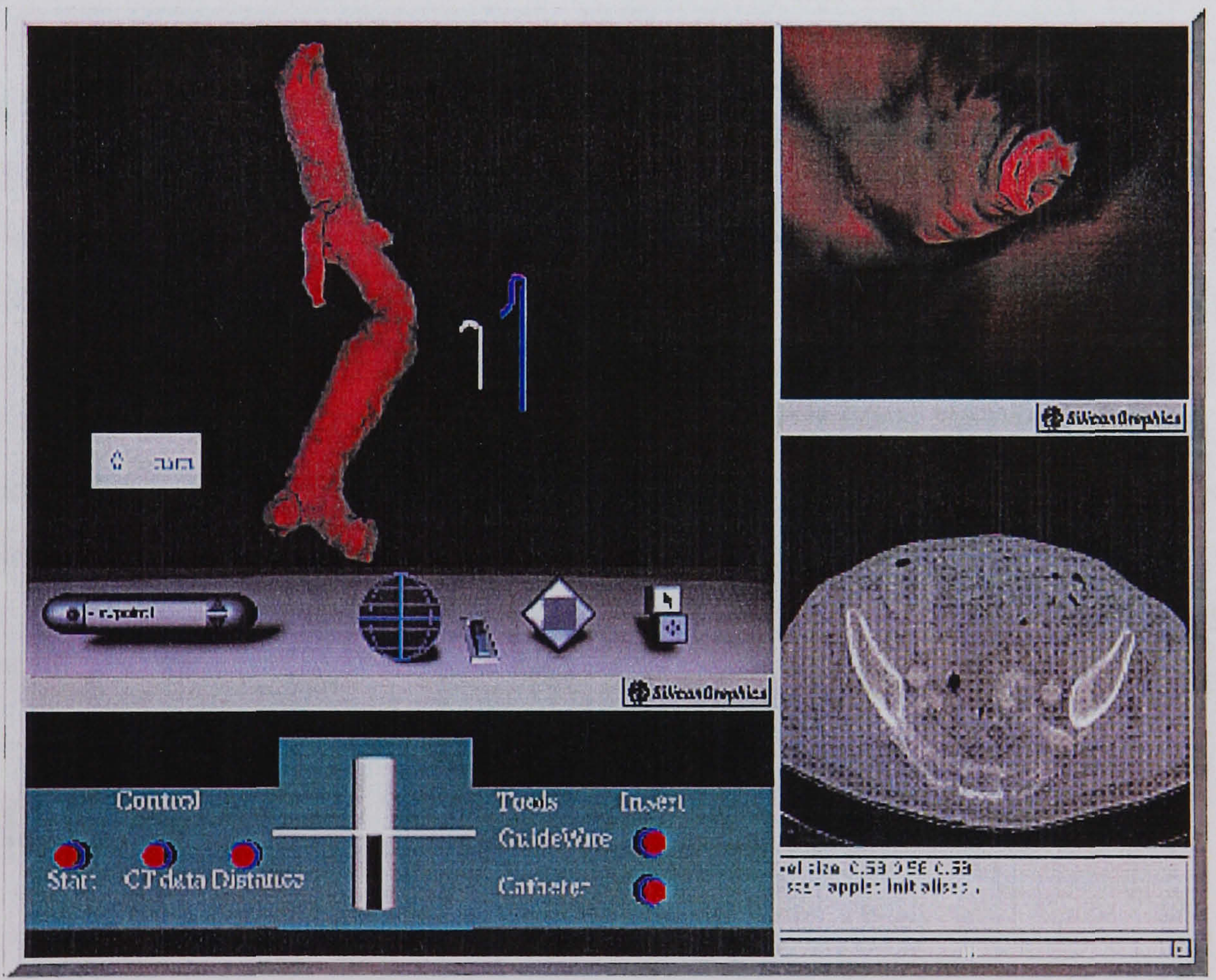
Because we adopt a server-based approach, communication becomes vital for our application. Currently, the communication between the server and the client is done over the "best effort" IP protocol. However, because of the isochronous nature of our application, a quality of service (QoS) protocol would better serve our application. Thus, when ATM is more popularly used on the Internet, performance will improve.

Meanwhile, our application keeps communication to a minimum.

In order to locate our surgical training application among other web-based applications, we review our solution in terms of the three proposed aspects of web-based classification: HCI, communication and application. In terms of HCI, we provide virtual object manipulation to interact with the training environment- details are in chapter (5). From the perspective of the 'communication axis', our solution is a single-user system. Finally, in terms of the 'application axis', our training system provides modelling of anatomy- geometric modelling- as well as behaviour modelling of the surgical tools.

Chapter 5

WebSTer user interface



5.1 Introduction

Barfield [33] presents the components of the virtual environment from a different perspective to the one we presented in section (2.1.1). According to him the virtual environment consists of hardware and software to model and render images, a physical interface to present the environment to the human senses and a logical interface to manipulate the environment attributes. It is the last component in which we are interested in this chapter. This component has been long known as the graphical user interface (GUI) in the computer graphics field. This component can be described as the software support of the physical input/output interface. It can be one-to-one representation of the functionality and geometry of the actual device, or it can be an abstraction or adaptation of this device.

User interfaces can be either generic or task-dependent. We use generic interfaces in most desktop applications: menus are used for selection and buttons are used to trigger actions, this is a simple and abstract approach. However, this type of interface is not able to support more specialised tasks- e.g. performing surgery. Thus we have the role of custom interfaces with specialised tools to suit the requirement of the task.

In addition to being task-sensitive, user interfaces are user-sensitive. In the medical field, the same patient data and records are treated differently by nurses, clinicians and physicians. Systems which target more than one group of users with different characteristics usually provide separate interfaces for each group or at least allow visibility of options according to the group they belong to.

Finally, in virtual environments, user interfaces depend on whether the environment is immersive or desktop. This is due to the fact that in immersive environments, the user is not able to use the usual keyboard and mouse devices. In addition, extending the desktop metaphor in 3D becomes difficult. In this chapter we will look closely into the interaction techniques used in both desktop and immersive virtual environments and the suitability of each technique- section (5.2). Then we will list the requirements of the user interface of our application- section(5.3). In section (5.4), we will describe the user interface of our surgical training system. Finally, section (5.6) will discuss our solution based on the previously listed requirements.

5.2 Interactions

5.2.1 Types

Virtual environments have been developed for a variety of purposes, from furniture layout to fighting battles and performing surgery. Although the purposes of these environments are different, they share a basic number of interaction classes. These classes are: viewpoint manipulation, object manipulation and application control.

Viewpoint manipulation

Viewpoint manipulation is usually done during navigation and exploration tasks. It involves the control of the position, orientation, field of view and parameters of the camera in the scene. During navigation in 3D worlds, users tend to lose their sense of direction; they either do not know where they are or they cannot get to a place previously visited, or they do not know how to get to their target. Being lost in a world is usually due to two factors: the frame reference of the viewpoint manipulation and the visual feedback provided. The latter describes the way our bodies operate which is "we perceive in order to move, but we also move in order to perceive" [81].

The navigation frame reference can be the observer himself, in what is known as the *egocentric* frame reference. This frame of reference gives the natural feeling of viewing the world from the observer's own eye. On the other hand, in the *exocentric* frame reference, the viewpoint is fixed to a point and the user manipulates the world to get an outside view of the environment. Both modes are useful and cover different aspects of navigation. Thus, the egocentric frame reference- such as immersed perspective or flying vehicle- is useful for exploring the details of the environment. Conversely, the exocentric frame reference- such as bird's eye- is useful in getting the overall picture. In order to make use of the exocentric views, a pursuit tracking mechanism- i.e presenting both the target and the current position- is essential. On the other hand, egocentric views are better utilised when supported by compensatory tracking mechanism, where the magnitude and direction of the divergence from the target is indicated [105, 63].

There are a number of techniques that have been adopted to solve the problem of getting lost in an environment. One way is to use trailblazing, which is to drop marks during navigation to find previous locations. Another way is to use maps to plan your route [63].

Object manipulation

Changing the position and orientation, grabbing and other forms of interaction with the objects of the environment are usually carried out by the user to accomplish tasks. The manipulation of objects consists of two parts; the first part is to locate the objects, the second is to apply action to them. Both parts of this manipulation are not so intuitive in current virtual environments. For example, locating objects in the environment is difficult; users tend to lose the objects in the same way that they lose their sense of direction. In addition, applying actions to the object is hard, because of the lack of haptic feedback- which we heavily depend on when grabbing objects. The degree of difficulty is multiplied when using 2D devices to manipulate 3D objects. In this case graphics applications use bounding boxes and virtual track-balls to manipulate 3D objects with a 2D mouse.

A number of techniques have been used to solve the problem of locating objects. In [123] it has been suggested to use the proprioception ability- the sense of position and orientation of one's own body and limbs- to locate objects in immersive environments. Thus, the user would locate the objects with respect to his own body- e.g. above his head or in his hand. The objects move with the user, keeping the same relative position so they can be easily found when needed. Another method is to scale down the world to allow the user to pick the desired object and manipulate it, then scale everything up again. Other techniques use the gaze, hand or wand direction to select objects from distance, however these methods require unobstructed view of the object.

To apply actions to the object, one can use abstract tools such as sliders, buttons and dials. A more intuitive method of manipulation is to use virtual objects which correspond to real life tools- e.g. virtual hand. A more sophisticated manipulation system would track the user hand to apply actions on objects.

Application control

In order to control the environment parameters that are not part of the simulated world- e.g. rendering mode, speed of travel and options displayed- the user needs to manipulate application controllers. These affect the way the user interface corresponds to the user actions. Buttons, menus and sliders have been used to facilitate this feature to the user.

5.2.2 Techniques

- **Cursors**

  The cursor paradigm has been extended from the 2D desktop interaction to be used in 3D worlds. Three dimensional cursors can be used in a similar fashion as the 2D ones to select objects. However, because of the extra depth dimension, the cursor functionality and shape are adapted to the 3D environment. For example, the laser cursor has been designed to select objects from a distance by firing a ray from the cursor position in the direction of the cursor arrow. The object which is first hit by this ray is selected. Similarly, a magnetic cursor can be used to select a group of objects within the magnetic field of the cursor [110]. The jack and skitter are another example of cursors which adapted new shapes to suit the 3D world [81].

- **Menus**

  Menus are another example of interaction techniques that have been extended from the 2D desktop. Nevertheless, their drawbacks have been greater than their benefits in 3D worlds. Floating menus tend to get lost in the world and when they are fixed, they occlude other objects in the environment. The proprioception method- mentioned earlier- can be used to locate menus, while keeping them out of sight when they are not needed. The second problem with menus is that options can be difficult to select because of the extra depth dimension in 3D worlds. In this case laser cursors can be used to activate options.

- **Speech**

  Humans are used to interact with each other using vocal commands. Therefore, some virtual environments include speech recognition components in their environment to facilitate this interaction technique.

- **Gesture**

  Gestures usually accompany speech commands to illustrate a direction, shape, size or some other aspects of the spoken context. Furthermore, gesture as well as posture are used to interact with objects of the world. For instance, we lean forward to examine objects more closely- like zoom-in in desktop applications. Another example is how we use our arm and index finger to show direction or to select an object. In addition, we put our hand behind our ear to indicate that we did not hear the spoken command and to ask for repetition. It is these types of day-to-day gestures that need to be incorporated into the virtual environment.

- **Tools and Widgets**

  In [81] tools are differentiated from widgets according to the metaphor they represent. Thus, tools represent metaphors of real life objects, while widgets are abstractions. Tools and widgets are useful methods of interaction because they can be designed to suit the needs of a particular application. Hence, they interpret the user input directly and apply it to the object. The number of attributes of a tool or a widget that the user is allowed to change depends on the DOF required by the applications and the I/O devices. Nevertheless, it is essential to be careful not to clutter the tool with a lot of information that cannot be handled by the user [187].

5.3 WebSTer requirements

The user interface of our type of application carries major importance, because one of the main skills that we would like to transfer through our training system is eye/hand coordination skill. Hence, as mentioned earlier in the training requirement- page (40)- the visual/ kinesthetic interaction of the simulator should be similar to real life. Secondly, the interface should provide the force feedback that will allow building psychomotor skills. Finally, the interface represents the pictorial media that will be used to build the representational skill. With these three targets in mind, we can establish the requirements of our interface.

In section (2.1.2) we have found from the experiments reported by others to evaluate the suitability of VR in training, that a bird's-eye view point has helped in the transfer of the representation skill. Thus, we need to adopt an exocentric view point frame reference in our interface to facilitate the acquisition of this skill.

As we reported in the discussion of the previous chapter, we are not able to provide physical input devices that mimic the real life surgical tools or output devices that would support force feedback because of the web restrictions. Thus, our interface will need to provide a virtual tool to substitute for these features. This virtual tool will be an abstraction of the various catheters and guidewires used in reality. Visual or audio stimuli can be used to substitute for the force feedback as we saw in other medical simulators reported in chapter (2). Therefore, when designing this tool three considerations should be taken into account:

- The virtual tool should represent the geometry of the surgical tools used- which are long tubular devices with different tip shapes.

- The virtual tool must allow manipulation in the same way a surgical tool would be manipulated in real life. Hence, the user can push or pull the device in a one-dimensional translation and twist it clockwise or counter-clockwise in a rotation around its longitudinal axis.

- The virtual tool must also provide visual or audio force feedback to substitute for the haptic feedback.

Another factor considered when designing our interface is the experience of our users. Experienced radiologists have been trained to perceive anatomy using 2D slices and they tend to underuse 3D constructions. On the contrary radiology trainees do not have this skill. As we saw in other medical virtual environments- reported in chapter (2)- providing the traditional 2D alongside the new 3D methods is useful and forms a link that assists users. Finally, the user interface needs to provide control widgets to choose and activate the training options.

5.4 WebSTer interface

The user is able to interact with the training environment via a set of VRML worlds and Java applets that are downloaded with the WebSTer HTML page. The advantage of using VRML in the user interface is that it takes care of rendering the scene. In addition, it provides a number of sensors that can be used to create custom tools and widgets. There are three VRML worlds in our interface. One VRML world, called the *global viewer* provides an exocentric viewpoint of the training environment. Meanwhile, the second VRML world, called the *local viewer*, provides an egocentric view of the training environment. Finally, the third world is the *control panel.* Together with the VRML worlds, two Java applets are embedded in the HTML page. One Java applet acts as an image display screen, which is called the *CT scan applet.* The second applet is responsible for communication with the server- via sockets and with the VRML worlds- via the EAI; this applet will be called the *communication applet.* Figure (5.1) shows the user interface layout of WebSTer. In the next few sections we will explore the components of the interface in more detail.

5.4.1 Global viewer

The training environment consists of a surface representation of the aorta vessel extracted from CT scans of patients suffering from abdominal aortic aneurysm- as will be described in chapter (6). Two surgical tools are modelled in the environment: the catheter and the guide wire. The global viewer provides an exocentric view that will show the whole vessel and the surgical tools. This viewer utilises the "Examiner viewer" widgets provided by default with the VRML browser to allow the user to explore the 3D model from different angles. This viewer provides the pursuit tracking

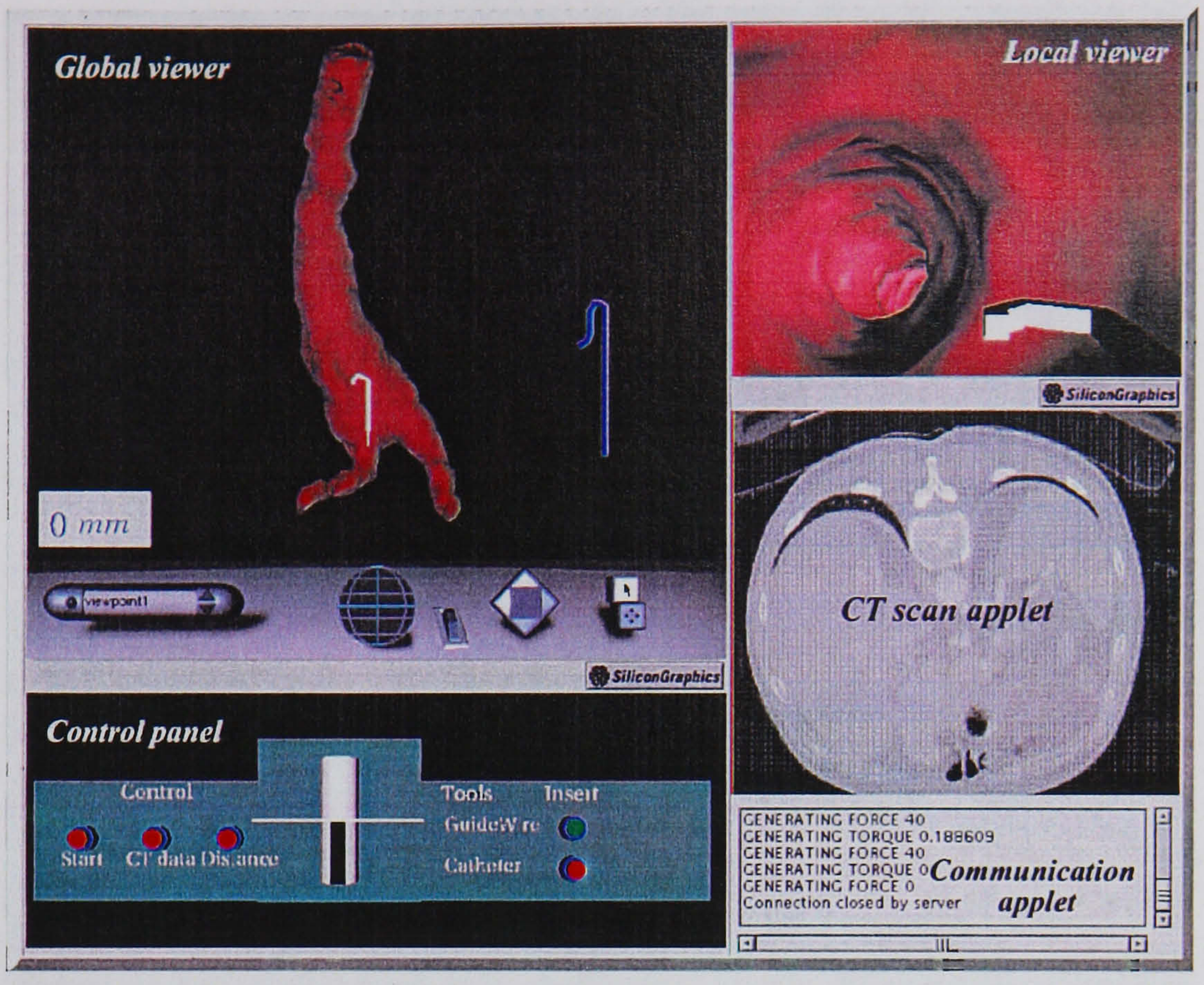


Figure 5.1: WebSTer user interface

mechanism to assist the trainee in knowing where (s)he is and where the target is. Thus, the current position of the surgical tool is always shown on this viewer. The blood vessel is made transparent enough to see the tool inside it. In addition, because the viewer shows the whole blood vessel, the position of the aneurysm- the target- can always be compared to the surgical tool position. Furthermore, the route traversed by the surgical tool is tracked by a thin line. This facility helps the user visualize the route taken so far. Because our simulator models the tip of the surgical tool only, this feature helps in reducing the effect of perceiving the tool as hanging in the air and links it to the insertion point. Figure (5.2) shows the global view of the aorta with a J-shaped guide wire tracked by the viewer using thin lines.

In addition to the environment objects, the global viewer provides two other widgets. The first is a cursor that is used to sweep the aorta surface to measure distances. The cursor is attached to a plane sensor that allows it to move in a two-dimensional plane in front of the blood vessel. When this widget is activated- from the control panel- moving the cursor will highlight the area covered by the motion- as shown in the top left window of the figure (5.2). The distance covered in the direction of motion is then reported on a text screen panel at the bottom left corner of the global viewer. Using this method of measuring distance will allow the measurement of oblique as well as orthogonal distances, since blood vessels are

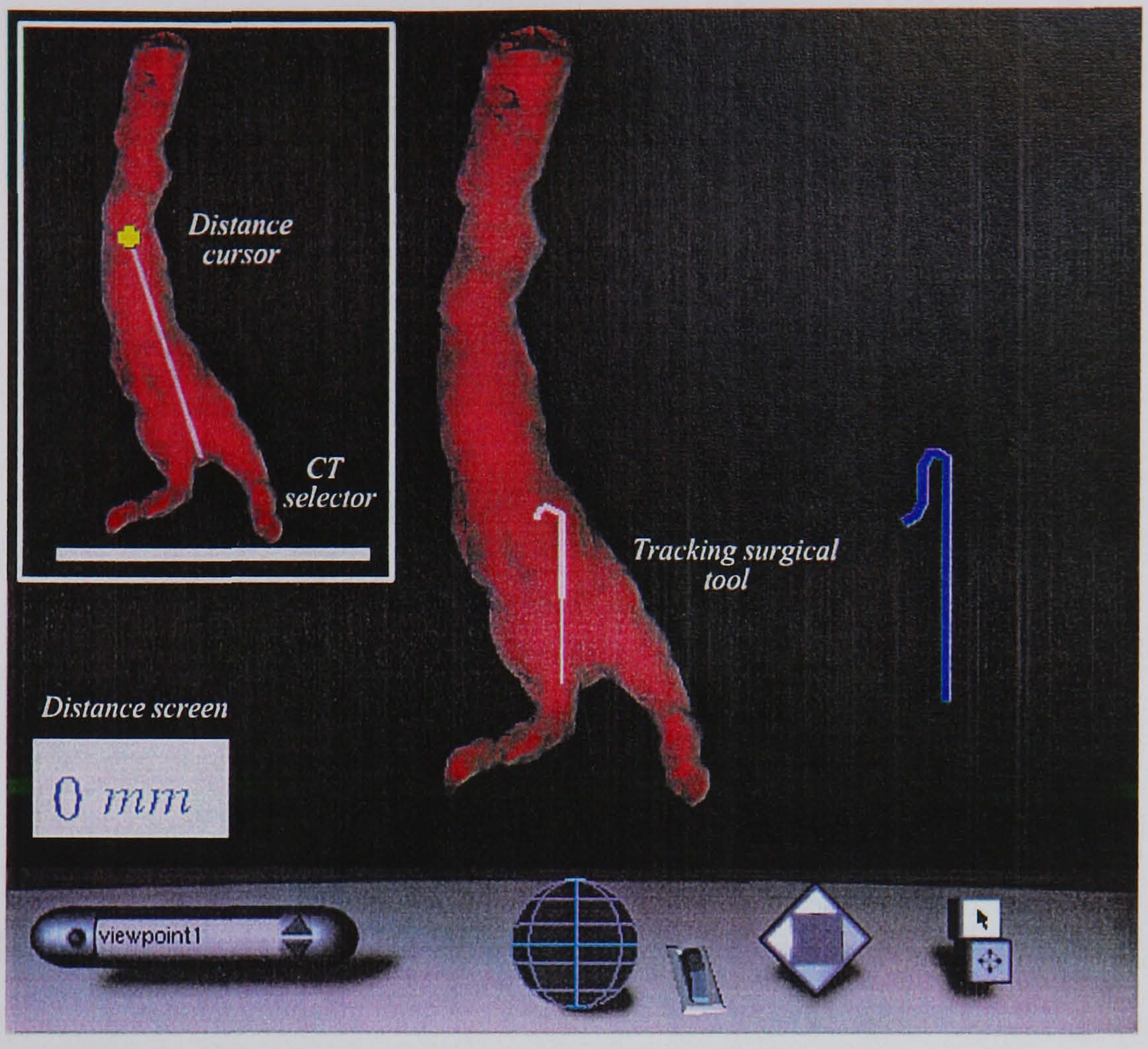


Figure 5.2: Global viewer

not always straight. This tool can be used by the trainee to measure the aneurysm neck, bifurcation distance and diameter of the vessel, which are essential to decide the suitability of the procedure and to plan it.

The second widget provided in the global viewer is a selector bar that can be moved along the longitudinal axis of the blood vessel. This selector is linked to the CT scan applet and will be further described there- section (5.4.4).

5.4.2 Local viewer

In this viewer, the viewpoint is placed in the current position of the tip of the surgical tool with the current orientation, providing an endoscopic view of the vessel- see Figure (5.3). The user is not allowed to manipulate the view, but rather it is updated by the simulation according to the current position of the surgical tool. The tip of the surgical tool is shown in the view with the current orientation of the tool. This viewer can assist the trainee when diving into branches to make sure that the tip of the tool is pointing in the direction of the branch.

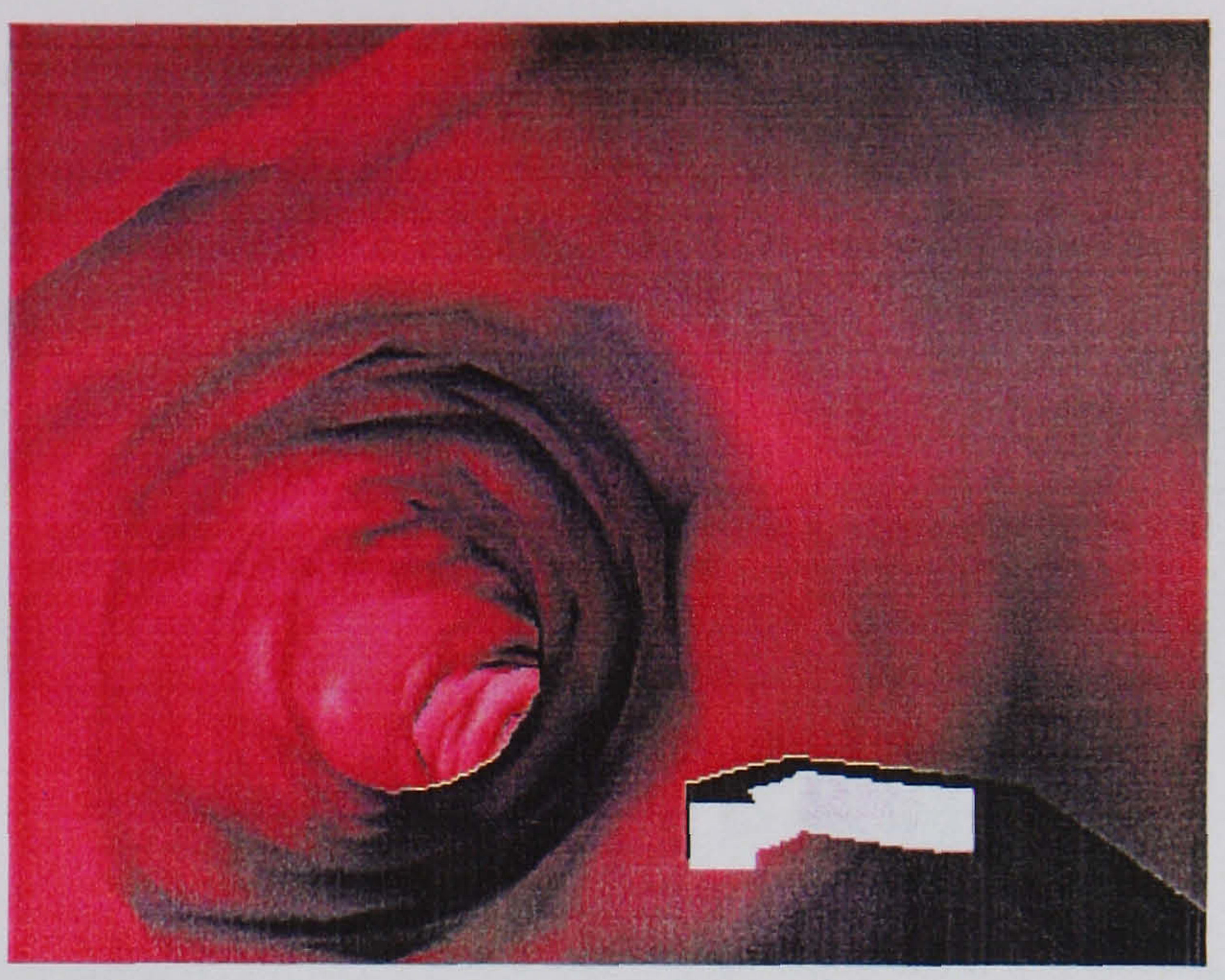


Figure 5.3: Local viewer

5.4.3 Control panel

This VRML world provides the user with buttons to activate the widgets in the global viewer. These widgets are hidden by default to avoid cluttering the global view of the vessel. When the user presses the button of a particular widget, the transparency attribute of the widget is altered. In the case of the distance measurement widget, the button is a three-state button to allow activation, standby and deactivation of the tool. In the active mode, if the tool is moved, the distance travelled is highlighted. Meanwhile, in the standby mode, the widget can be moved to the initial desired position without highlighting the motion. Finally, in the inactive mode, the tool is transparent.

In addition to activating widgets, the control panel is used to activate and deactivate the surgical tools- i.e. guide wire and catheter. As mentioned in chapter (3) the physician uses a combination of both tools to manoeuvre inside the vessel. Each surgical tool is provided by a three-state button to control it. Initially, all the tools are outside the vessel and the buttons are inactive. When the user clicks the button of some tool, the tool is moved to the insertion point position- in the iliac artery- to start the simulation, and the button is placed in standby mode. By clicking on the tool button again, the tool is activated. A tool that is in the active mode means that forces and torques applied by the user will be redirected to this tool and the simulator will calculate its behaviour accordingly. The control panel is developed such that at any time only one surgical tool is active, while any number of tools can be in standby mode. The application also assumes that catheters cannot be inserted without a guide wire inserted first, hence it will not allow such action. Figure (5.4) shows the control panel VRML world.

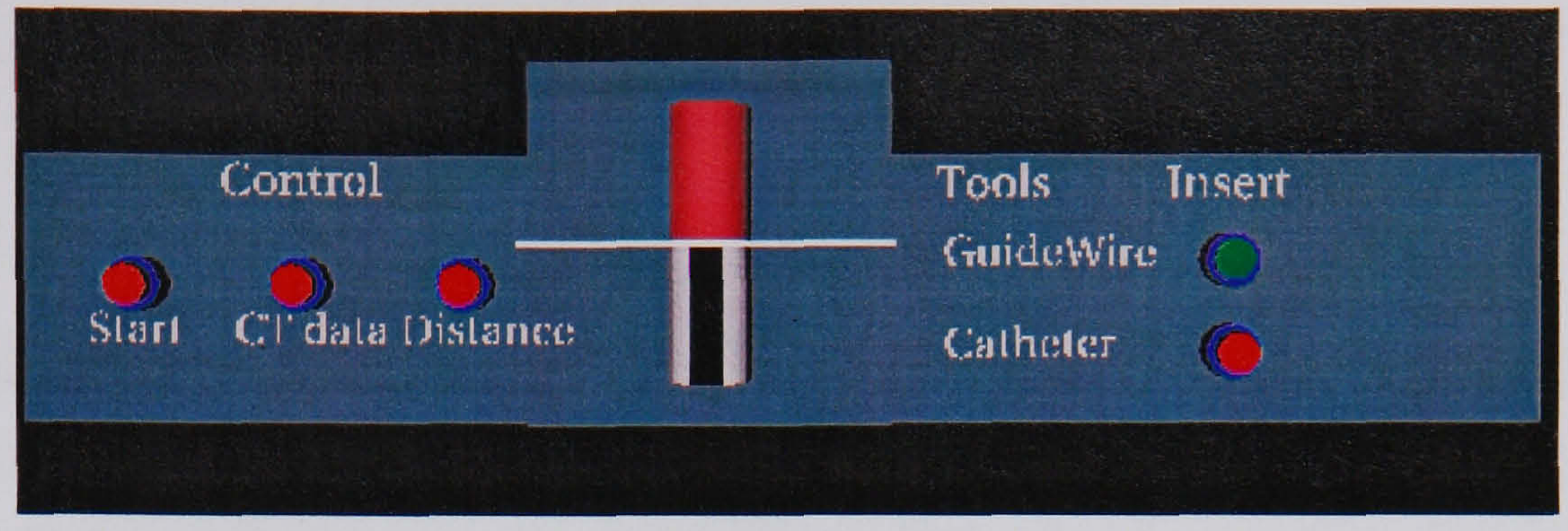


Figure 5.4: Control panel world

The panel also provides a virtual tool that corresponds to the real life surgical tool and meets the requirements listed in the previous section. The virtual scope that we provide is a cylinder with two parts. On the top part, the strength of the collision forces is translated into colour intensity and is reflected in the colour attribute of this part. The second half of the virtual scope is texture mapped with colour stripes to give a sense of the current rotation of the tool. The tool is placed on a pad that defines the extent of its motion. This allows the control of the amount of force and rotation applied to the tool. The vertical motion of this tool is interpreted as pushing or pulling forces. The amount of force applied increases as the user gets further from the centre. Meanwhile, the horizontal motion is interpreted as rotation clockwise or counter clockwise. The rotation angle increases as the user gets further from the centre. Using this virtual tool the 2D motion of the mouse device can be interpreted according to the need of our application. A different mapping needs to be used if the user utilises 3D devices. However, since the tool is implemented as a VRML prototype node- a user defined node with certain Events In,Out and fields- there is no need to modify the application in any way. The new mapping can be implemented in a new prototype node that gives the same interface. Thus, as far as the application is concerned, force and rotation angle are being sent from the interface no matter what device is used or how it is mapped.

Another button is provided by the control panel to activate and deactivate the simulator timer. When this timer is active, time messages are sent every time cycle to request updates from the simulator on the server side. Initially this timer was implemented as a VRML "TimeSensor"- a node which generates "Events Out" as time passes- that was included in the global viewer. However, this timer does not produce real time events- e.g. the timer fails to produce an accurate number of events for a frame rate greater than 10 frames/second on an SGI O^2 (180 MHz, 96 Mbyte memory) machine. To circumvent this problem, we created a Java thread out of the communication applet to act as a timer. By assigning the thread maximum priority, the Java timer gave better performance- i.e. it can produce accurate events for 25 frames/second- on the same machine.

5.4.4 CT scan applet

This Java applet is used to display the CT slices that were used to construct the 3D geometry. The selector widget provided in the global viewer links between a position on the 3D model of the blood vessel and the current displayed slice. For experienced radiologists, this tool would act as a link between the traditional method of viewing anatomy as slices and the 3D representation provided. On the other hand, for a trainee, this tool would help to establish the skill of transforming the 2D slices to 3D mental structures- representational skill.

In an earlier version of the interface, the images were displayed as a texture map on a box in the VRML world. However, as the number of texture maps increased in the HTML page, we noticed that the VRML renderer (SGI Cosmoplayer 1.0.2) stopped displaying them. Consequently, we used a Java applet to display the images instead. The first time the tool is used, image display is slow; however, once they are loaded in memory they are interactively displayed.

5.4.5 Communication applet

The user interface of this applet is simple. A text area is used to output messages to the user about the state of the communication between the client and the server. Textual feedback is also provided for user action, e.g. when activating or deactivating surgical tools. Error messages are printed in this area to warn the user of any illegal actions. We have also experimented with adding audio output messages to guide the user through the session and to report incorrect manipulations using the Java sound capabilities.

The communication applet grabs pointers to the three VRML worlds using EAI functions. These pointers are used to get access to nodes and fields in the VRML world. The fields can be updated- in case of eventIn's- or detected for changes- in case of eventOut's.

5.5 Discussion

Our user interface provides both types of viewpoint frame reference- exocentric and egocentric- in order to support navigation. VRML does not allow multiple viewpoint display of a world. To avoid this problem, we duplicated the training environment in two different worlds and coordinated between them using the EAI. Thus, when an update is received for the surgical tool from the server, the communication applet will update the position and orientation of the viewpoint of the local viewer, in addition to updating the transformation of the surgical tool in the global viewer- refer to appendix (A). This solution unavoidably doubles the rendering time of the application.

The web limitations have hindered certain features in the simulator, but we have adapted our user interface to compensate for these features. Providing visual feedback on the virtual scope is probably cluttering the tool with more information than desirable. However, we have experimented with providing audio feedback, where the intensity of the sound clip is controlled by the strength of collision. VRML sound clips support such a display with intensity control. Nevertheless, we were faced with a similar situation as with texture maps; as more than one VRML world is embedded in the HTML page, the sound clips are not displayed any more. This might be because CosmoPlayer does not support multiple instances of audio (we tested on SGI Cosmoplayer 1.0.2 and PC Cosmoplayer 2.1). Alternatively, we can use Java to display sound clips; however Java does not control the intensity of the played clip. Hence, we have simply added the Java sound clip to alarm the user that collision has happened, while keeping the visual feedback in the virtual scope.

Chapter 6

WebSTer modelling

6.1 Introduction

As mentioned in section (2.1.1) one component of building a virtual environment (VE) is to create a model of it. This model contains objects and actors within this environment. In a virtual environment, objects are expected to respond to the actors in the environment, consequently the process of modelling an environment consists of two parts. First, geometric modelling, where the shape of the object is represented. Secondly, behaviour modelling which represents the motion and the reactions of the object.

In general, geometric modelling can be divided into either surface representation or volume representation- Foley, van Dam and et.al. is a good reference to the different techniques used to model geometry [66]. The choice of which method to use in modelling the geometry of an object depends on the criteria of the application. Usually, in virtual environments the speed of display and interactive manipulation are essential issues. For medical virtual environments, accuracy of the geometry is an added requirement. In addition, geometry representation in virtual environments is affected by the behaviour of the object [31]. For example, in modelling a human arm, because moving the shoulder will in effect move the rest of the arm, it is sensible to use hierarchical geometric representation of the different links in order to simplify behaviour modelling.

Objects and actors in the VE are described by a characteristic vector which is a collection of different attributes of the objects- e.g. orientation, position, velocity, colour and energy.. etc. Objects in the environment interact with each other as well as with actors in the environment. As a result of this interaction the object attributes change. Such change in the object's attributes is known as behaviour. It is useful to classify the different levels of behaviour. The lowest level of behaviour- level 1- will cause direct change of one of the object attributes, for instance change the position of the hand. A higher level of behaviour- level 2- would change an attribute over time to achieve an action- e.g. changing the position of the hand to reach an object. Moving to a higher level of behaviour- level 3- we notice that it is composed of a group of actions to achieve a task. For instance, move the hand, reach a cup and raise the cup up are actions taken to achieve drinking. Finally, the highest level of behaviour- level 4- is to make a decision about which task to perform next- e.g. decide whether to drink or eat [145].

Hence, we can classify objects in a virtual environment into three classes: static, Newtonian or intelligent [145]. Static objects exhibit no change in attributes, consequently they are only geometrically modelled. Meanwhile, Newtonian objects exhibit behaviours of level 1 and 2, which are based on some scientific law. For these types of objects, *physically-based modelling*- which will be described later in detail- is usually utilised to represent their behaviour. Finally, intelligent objects are capable of higher levels of behaviour- level 3 and 4. In order to model their intelligent behaviour, *behavioural animation* techniques are used. These techniques are based on the idea of sensing the environment and using some form of intelligent techniques- e.g. rule-based system or neural networks- to decide on the action taken [31].

In this chapter we are concerned with issues related to geometric and behaviour modelling of our surgical training system. In the next section, we will introduce the different stages of a systematic modelling process. The structure of the rest of the chapter is based on these stages. In section (6.3), the abstraction stage will help us answer questions like: What type of objects are we dealing with? and what kind of behaviour do they exhibit? This stage will produce a set of modelling requirements for our training system. Then in section (6.4), the representation stage will decide based on these requirements the type of modelling techniques needed. Section (6.5) will present background information for these techniques. Then, section (6.6) and (6.7) will describe the design and implementation of the modelling aspects of our training system. Finally, we will conclude the chapter with the usual discussion, which evaluates our solution.

6.2 Creating a model

Among all the components of creating a virtual environment, modelling is the most application-dependent activity. Furthermore, virtual environments usually deal with a wide range of objects with various geometric and behaviour properties. Thus it would be useful to create models which can be reused, extended or modified. This paradigm- known as component modelling- is becoming increasingly popular in the simulation and modelling field. There are two aspects that would help in achieving such models. One is to create models which exist independently of their environment, such that they can be removed and placed in a different environment without the need for further modification. The second issue, is to utilise a systematic method for the creation of the model together with a well-defined way of communication between models. This will reduce the effort required to reuse or modify the existing models.

To define a systematic way of creating models, we need to explore what we mean by a model. Lehman in [112] describes a model with the following collection of statements:

- A model is a representation of a theory.
- A model is an abstraction of key elements of a theory.
- A model is an application of a theory to a specific situation.

From Lehman's point of view, a computer program, a physical model and a mathematical model are all different representations- hence different models- of a theory.

This definition of a model contains all the elements of a model but in a vague sense. On the other hand, Barzel [35] presents a similar definition of a model but with sharp distinction between the different stages of a model. Barzel defines a model as "an implementation of a representation of an abstraction of a thing". Hence, building a model includes three stages. First, the creation of an abstract or conceptual model of the object, where a list of the properties of the object is prepared. Second, the abstraction is described in a formal and concise way. At this stage, the theory- as described by Lehman- explaining the modelled object is utilised to give the formal representation. Finally, the model is implemented in some media; such that a computer program or a physical model are different implementations of the representation. In addition to these stages, Barzel emphasises the importance of keeping a general goal for the modelling process to act as a guideline. Barzel's method- called the ARI model- represents a structured methodology for creating a model. Figure (6.1) shows the stages of this method - the figure is adapted from [35].

For example, consider modelling a heart valve for the purpose of visualizing the blood flow pattern through the valve. In the abstraction stage, geometric properties of the heart valve together with biomechanical properties of the blood are listed. Then a set of fluid dynamic equations- which represent the flow of blood within the boundary conditions of the valve- form the representation of the model. A computer program which solves these equations is an implementation of the model.

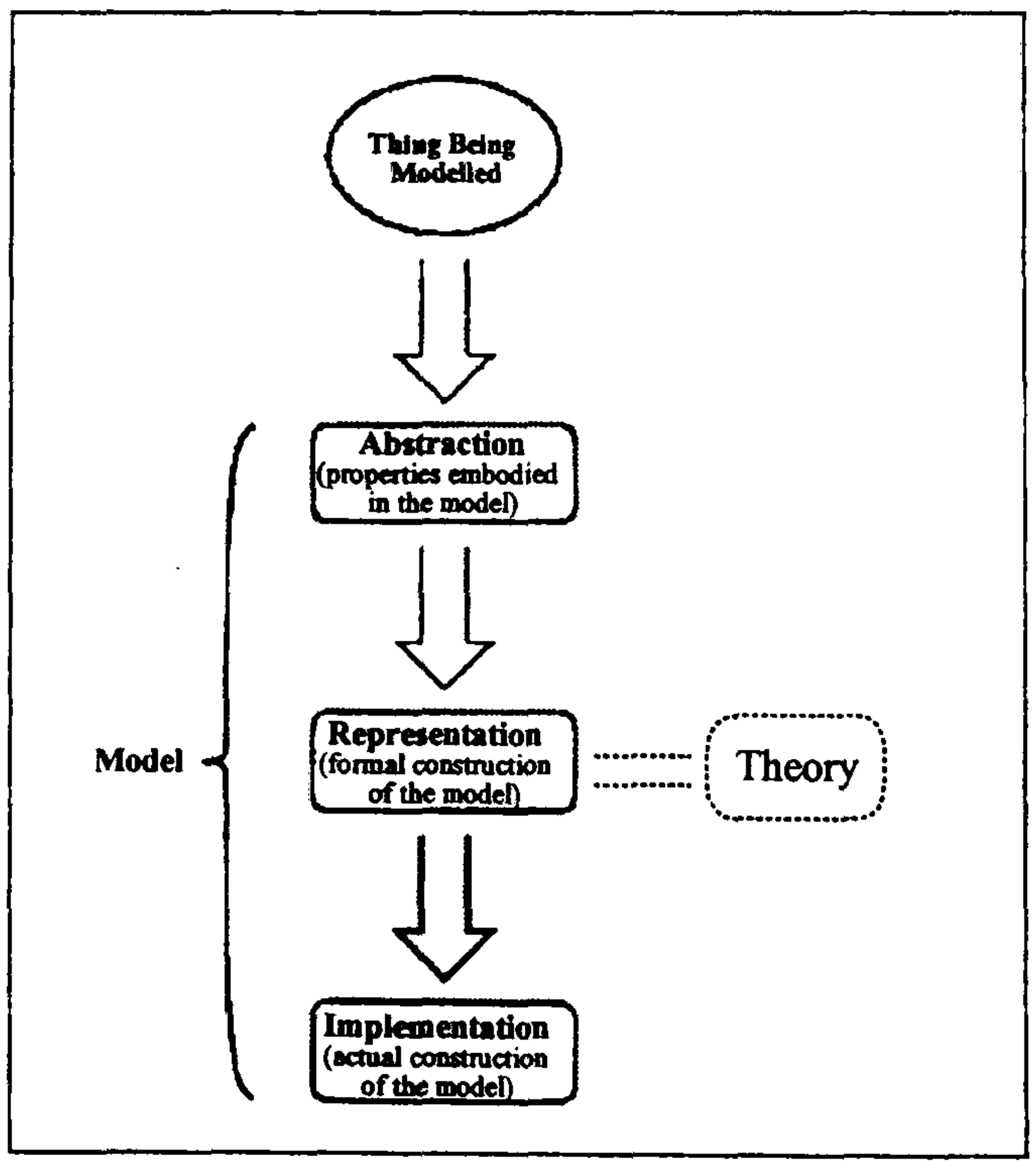


Figure 6.1: ARI structure of a model

Because of the clear separation between the stages of building a model, the ARI method has a number of advantages:

- The conceptual model includes all the properties of the object. Meanwhile, the representation of the object is a more restricted view of the object within some framework of interest. Hence, the existence of a conceptual model facilitates the expansion of the model, or the adaptation of a different representation.
- The separation of the conceptual model and the representation makes the model easily adaptable to other objects.
- The separation of the implementation from the representation facilitates the experimentation with different methods, and the exploitation of existing solutions in the model. For instance, in the heart valve example, we would be able to make use of different existing numerical solvers in the implementation without the need to modify the representation.

In the next few sections, we will apply these stages for the modelling of our surgical training system. The actual modelling process of this system combined a top-down and a bottom-up design approach. This was due to the feedback from radiologists in St. James's University Hospital, where our knowledge increases as we go along and consequently we adjust our model.

6.3 WebSTer modelling: Abstraction

Initially, it is essential to state a general goal for the modelling process which will be refined as the model develops. Our aim for modelling the endovascular procedure is to provide initial training for radiologists. For this purpose we are building a virtual environment where the training takes place. The content of this virtual environment consists of two components: the surgical tools- guide wire and catheter- and the blood vessel. The radiologist is an actor in this virtual environment. This actor is not visually represented in the environment; however their actions are modelled. Hence, in the abstraction stage we will consider the properties of the two components. These properties are classified into three types.

- Geometrical properties: describe the shape of the object.
- Physical properties: describe properties related to the manufacturing material of the object.

- Functional properties: describe logical units of the behaviour of the object which leads to the accomplishment of a task.

Section (6.3.1) lists the properties of the surgical tools- guide wire and catheter. Then, section (6.3.2) describe the blood vessel properties.

6.3.1 Characteristics of the surgical tools

6.3.1.1 Geometrical properties

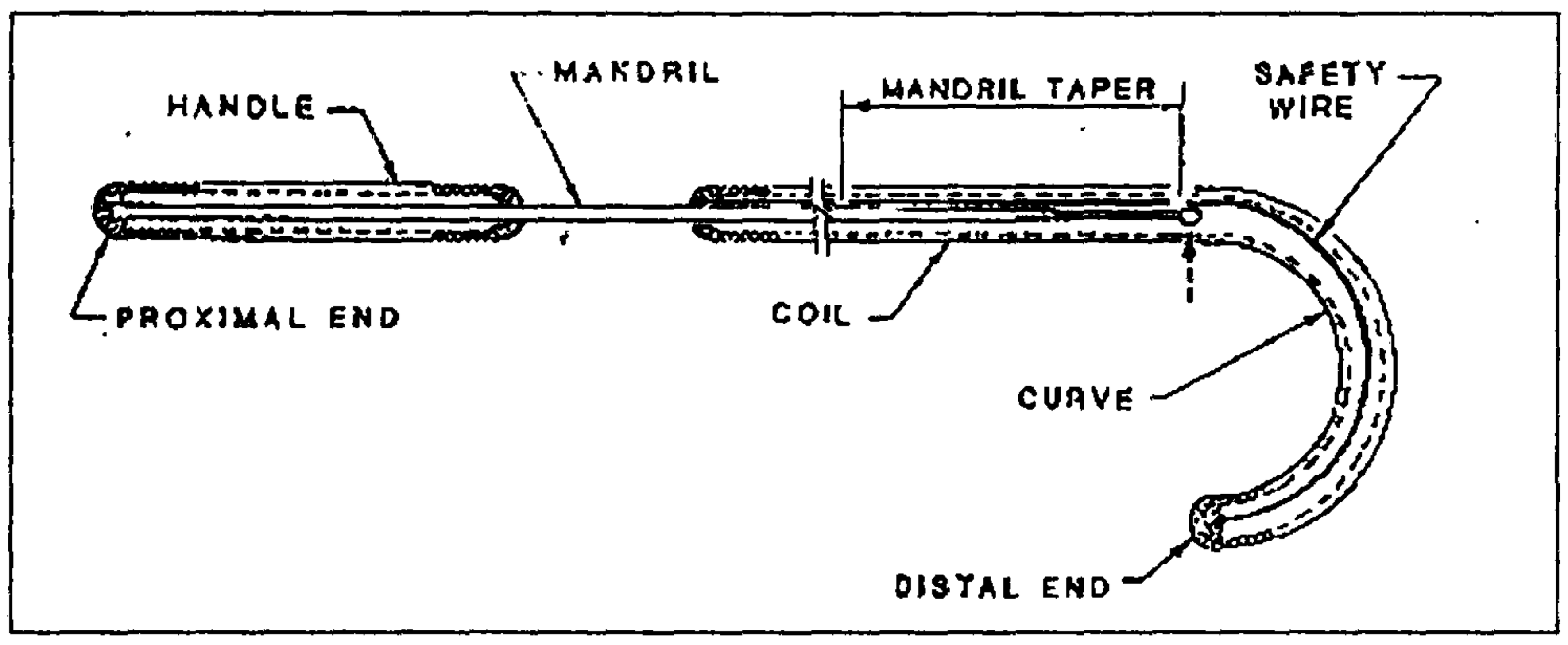


Figure 6.2: Structure of the J-shaped guide wire with movable core

A guide wire consists of two parts, a tip and a shaft. Usually the tip is more flexible than the shaft. However, the transition is usually gradual over some area. The shape of the tip differs according to the function of the guide wire- e.g. a J-shaped guide wire is used to dive into arterial branches. In addition, the construction of the tip varies. The conventional structure of the wire consists of a core, a safety wire and a spring coil. However, variations of this structure can include a short safety wire welded to the core, no spring coil, no safety wire or a movable core wire [153]. The diameter of the guide wire core can range from 0.24 to 0.81 mm. Figure (6.2) shows the shape of a J-guide wire. To specify the shape of the guide wire, you need to quantify the following parameters:

- Length of the guide wire.
- Diameter of the guide wire core.
- Length of the tip.
- Diameter of the tip.
- Shape of the tip.
- Length of the transition between the tip and the shaft.

A catheter is a hollow tool with different tip shapes. It can be geometrically specified by: the length and the diameter of the catheter as well as the tip shape. Figure (6.3) shows different tip shapes for the catheter.

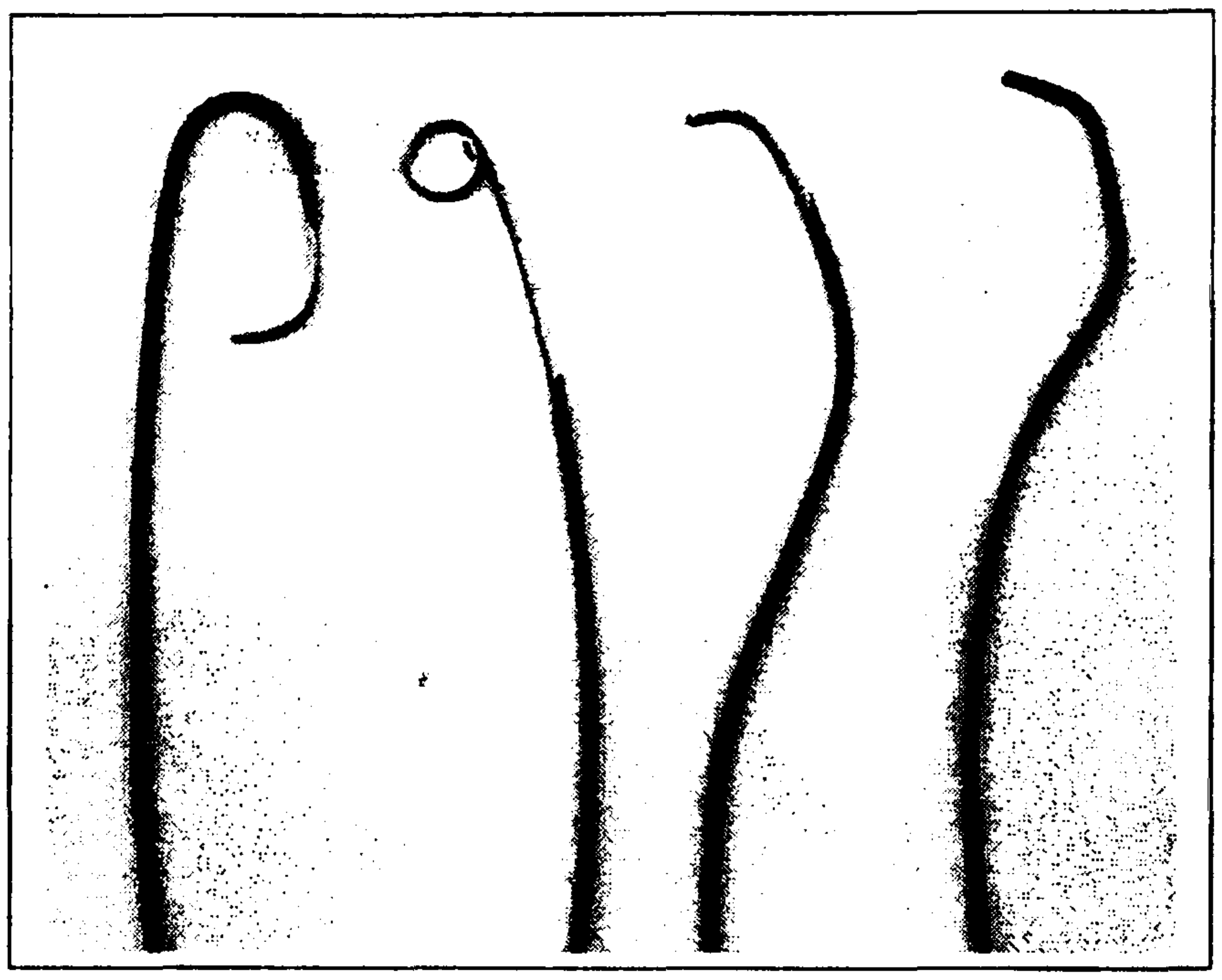


Figure 6.3: Different tip shapes for the catheter (picture extracted from Medi-Dyne web pages at http://www.catheter.com)

6.3.1.2 Physical properties

The material, structure and the diameter of the guide wire affect the mechanical properties exhibited by the guide wire. The guide wire can be made of steel or Titanium. In some cases they are coated with Teflon or a hydrophilic polymer material.

Mechanical properties such as stiffness, torsion and friction affect the performance of the guide wire in manoeuvring through tortuous structures. Experimentation and enhancements have been done over the last thirty years to improve the capabilities of the guide wire- e.g. [146, 144]. For instance, conventional guide wires hardly exhibit any torsion when subjected to torque. Coating the steel wire with Teflon for example has overcome this drawback. Sliding friction between the catheter and the guide wire was greatly reduced when coating was introduced [154].

A complete experimental study of the mechanical characteristics of different guide wires has been done by Schroder [153, 154]. Stiffness of the shaft, tip and transitional area has been reported by measuring forces which cause 45° bending of the wire. It was found that stiffness is proportional to the fourth power of the core diameter. In addition, the structure of the tip and the material affects the stiffness. Stiffness of the tip ranged from 0.02 to 0.33 ($E^{-6}Nm^2/°$).

The torsional strength of the wire was measured by restricting the motion of the wire to twisting and introducing different loads, while measuring the angle of rotation. Again a correlation between the core diameter and the torsional strength was noticed. Guide wires which are capable of twisting had a torsional strength that ranged from 0.42 to 105 ($E^{-6}Nm^2/^\circ$). Other properties such as kinking- the largest strain that the wire can tolerate without permanent deformation- has been measured. Sliding friction of the guide wire in the catheter, or of the catheter over the guide wire, has been reported as well.

In general, catheters are more stiff than guide wires. They are made of Teflon. However, we did not find equivalent studies for the catheter physical characteristics in the literature. Hence, we assume similar characteristics but with higher stiffness.

6.3.1.3 Functional properties

As mentioned before, the guide wire is used to facilitate catheterisation. Hence, the guide wire has a purely navigational function. On the other hand, a catheter has dual functionality. In addition to the navigational task, the catheter is used to introduce dye or to expand the balloon. The navigation function of both the catheter and the guide wire implies that these tools will respond to user-applied forces: longitudinal in the form of push and pull forces and torque in the form of twisting of the tool around the longitudinal axis.

In addition to this direct manipulation, the physician will use a combination of both tools and with the help of the vessel wall to manoeuvre the tools in branches or structures, During this type of manipulation, the catheter may slide over the guide wire/to follow the guide wire path. In this case the catheter shape changes to adapt to the guide wire shape. Conversely, the guide wire can slide inside catheters of different tip shapes to facilitate diving into smaller branches- e.g. J-shaped or sidewinder catheter- or to reverse the wire down to the bifurcation. In this case, the guide wire adapts to the catheter shape. Furthermore, the walls of the artery may be used to change the tool direction or to act as a pivot to support the tool during branch diving. Hence, the triple effect of user- tool- wall plays the major role in the navigation task.

The deployment functionality of the catheter involves the expansion of the graft under balloon inflation. The deformation of the graft is shown in Figure (6.4).

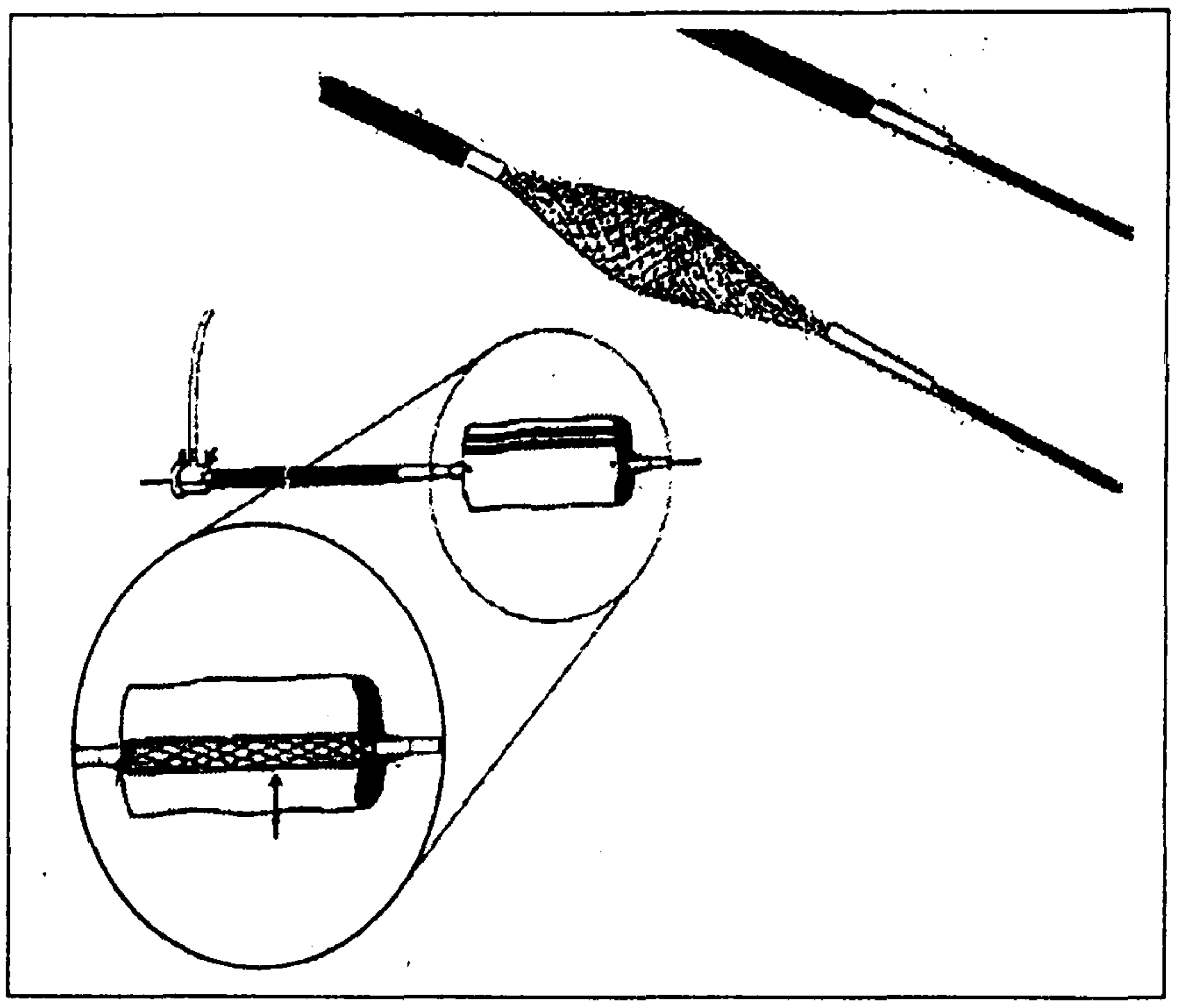


Figure 6.4: Deformation of the graft as the stent is deployed

6.3.2 Characteristics of the blood vessel

6.3.2.1 Geometrical properties

Blood vessels are thin-walled tubes with branching structures. The thickness of the vessel wall and the structure of this wall varies along the body. In addition, the diameter of the vessel varies along the vessel. In AAA endovascular procedures, the interest is focused on the aortic artery and its branches up to the kidney level. Figure (6.5) shows the arterial structure of interest. To specify the shape of the blood vessel, we require the following parameters:

- Diameter of the vessel.
- Thickness of the wall.
- Branching structures.

6.3.2.2 Physical properties

Blood vessels are made of soft tissues, which have non-linear elasticity properties. This non-linear elasticity increases as the vessel gets further from the heart. Hence, we notice changes in the elastic properties of the aortic vessel, which are due to changes in the material composition and the structure of the walls along the vessel. Normal blood vessels are subjected to biaxial loading due to blood pressure, together with longitudinal stretch and shear. To study the mechanical properties of the blood

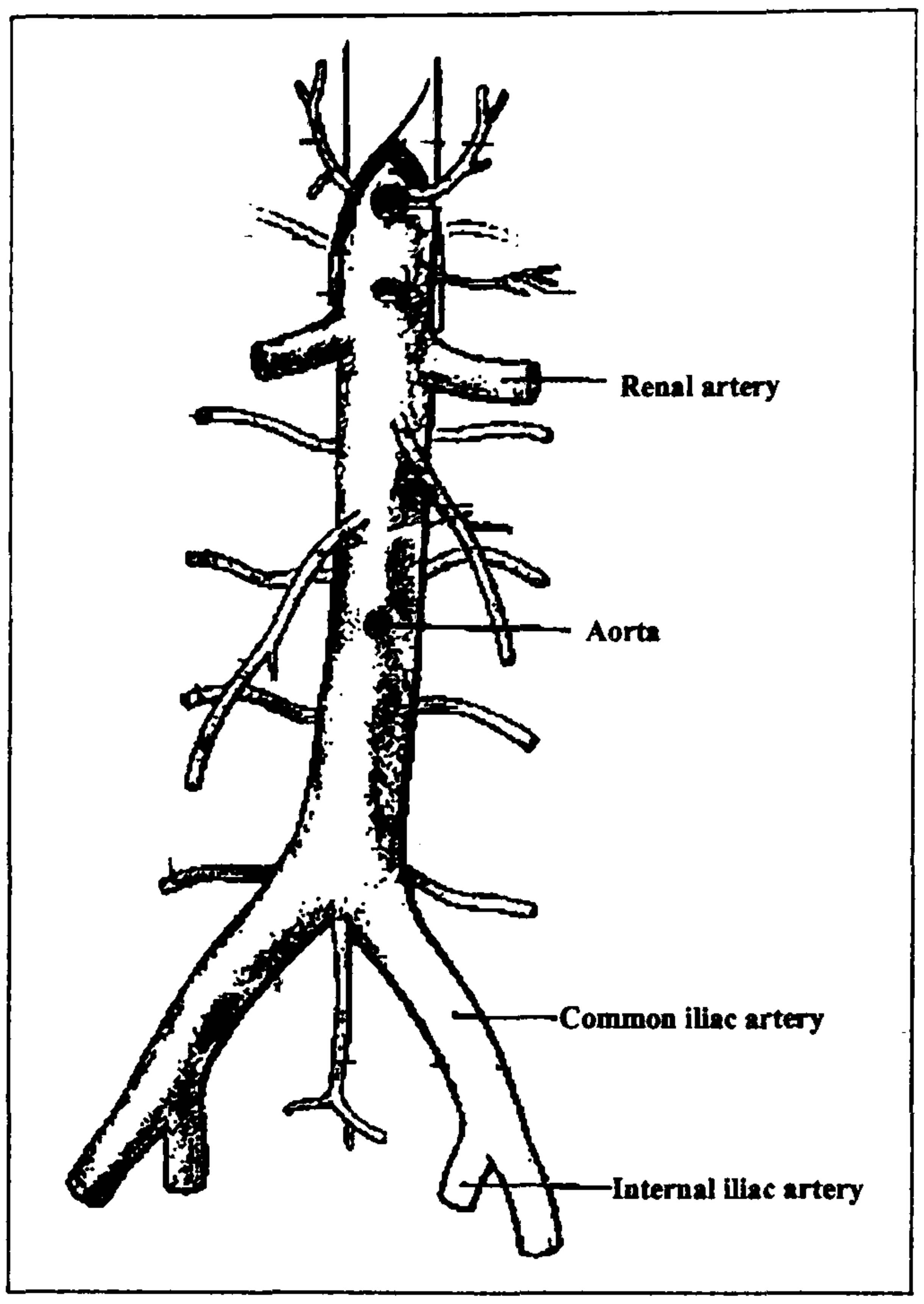


Figure 6.5: Arterial structure in AAA endovascular procedures

vessels under these loads, simplifications are sometimes necessary. Although blood vessels are three-dimensional structures, they can be simplified into two-dimensional curved membranes. In [68], Fung reports experiments and mathematical relationships which represents the torsion and stretching behaviour as well as mechanical properties of the blood vessels. Fung utilises the simplified 2D representation as well as the 3D representation whenever simplification is not appropriate. He also notes the changes in the blood vessel structure and mechanical properties as a result of diseases such as aneurysm and diabetics.

6.3.2.3 Functional properties

In the type of procedure we are interested in, the blood vessel holds three functional characteristics:

- The blood vessel wall deforms under the pressure of the blood flow.
- The blood vessel wall deforms as well as exerts some resistance force on the surgical tools as they collide with it.
- When collision is forceful, the vessel perforates causing bleeding.

6.3.3 Goal and assumptions

Now that we have introduced the properties of the components of our system, we need to state the goal of the modelling process. This goal and assumptions made in this section will affect the representation that we choose for our components.

The aim of modelling the surgical tools and the blood vessels is to simulate the behaviour of the tools under user interaction while navigating inside the vessels. We have made a set of assumptions to simplify the real situation. First, we assume that the wall of the vessel does not deform when the tools interact with it. Secondly, as we saw in the aneurysm modelling system in page (24), simulating blood flow requires high computations that are done off-line; hence we are not considering the effect of blood flow on the behaviour of the tools or the vessel wall. Thus, the surgical tools have been modelled in a void medium. The effect of gravitational forces on the modelled objects has been ignored. Finally, we are only interested in the navigational behaviour of the catheter- the effect of inflating the balloon in the catheter to deploy the stent and injecting dye are not considered.

Furthermore, we would like our model to coincide with the recommended characteristics of VE models. Hence, re-usability and modularity are two additional targets.

6.4 WebSTer modelling: Representation

Based on our goals and assumptions- reported in the previous section- we will develop the list of properties of our components- which resulted from the abstraction stage- into a formal geometrical and behaviour representations. Thus, in section (6.4.1) we will introduce the geometric representation of the surgical tools and the blood vessel that we adopt. In section (6.4.2) we will first examine the theoretical background that is necessary to understand the type of bevhaviour exhibited by the surgical tools. Then, we will report the representation we adopt to model the behaviour of the surgical tools.

6.4.1 Geometric representation

The surgical tools are represented as 3D models with square cross-section. The diameter of the tool corresponds to the width of the square cross-section. Furthermore, the model represents the tip of the tool with variation of tip shapes. As mentioned before, it is possible for the behaviour of the object to affect the geometric representation adopted. Since the ratio of the diameter of the surgical tool to the length of the tool is very small, it is possible to approximate the tool as a string- rather than a rod when calculating its behaviour- details will be reported in section (6.4.2.2). This will simplify the modelled equations and will reduce computations. In addition, it will reduce network communication as we recommended in the discussion section of the web aspects- section (4.7). Hence, during the processing of the tool behaviour, we use a one-dimensional model of the surgical tool. In this case, the tool is represented as a curve in 3D space. The 3D model of the tool utilises the VRML "Extrusion" node- which defines a surface in 3D space by describing a polygonal cross section that follows a path called the spine. The spine of this 3D model corresponds to the one-dimensional model.

We choose to use a surface representation for the blood vessel, in order to reduce the speed of display and interaction. Consequently, the volumetric CT scan data acquired from patients must be segmented to extract the aortic vessel with its branches. In CT scans the blood vessels have similar grey values as bone structures. Thus, conventional image processing techniques are not effective in extracting the aortic vessel from CT slices. Hence, we utilise a technique which uses a 3D deformable mesh to extract structures from images [42]. Initially, a 3D elliptic mesh is placed inside the vessel. Then image forces- which are generated from the distance to the nearest edge- and other constraint forces are used to expand the mesh towards the boundaries of the vessel. The mesh is adaptively refined to tolerate areas of the structure with fine detail- e.g. branches. The result of the segmentation is a polygonal representation of the vessel's surface, which is decimated to reduce the number of polygons. Figure (6.6) shows the process of modelling the geometry of the aortic vessel.

6.4.2 Behaviour representation

From the physical and functional abstraction of the surgical tools, we can see they are capable of twisting and bending- i.e. changing shape- when force is applied to them. When this force is under a certain limit, the tool restores its shape back when the force is removed. This implies that the surgical tools are Newtonian objects- see section (6.1)- and their behaviour follows elastic behaviour. Furthermore, based on the assumptions stated in section (6.3.3), we assume that the blood vessel is a static object- thus no behavioural modelling is required. Hence, in this section we will present the elastic theory supporting the behaviour of the surgical tool. Then

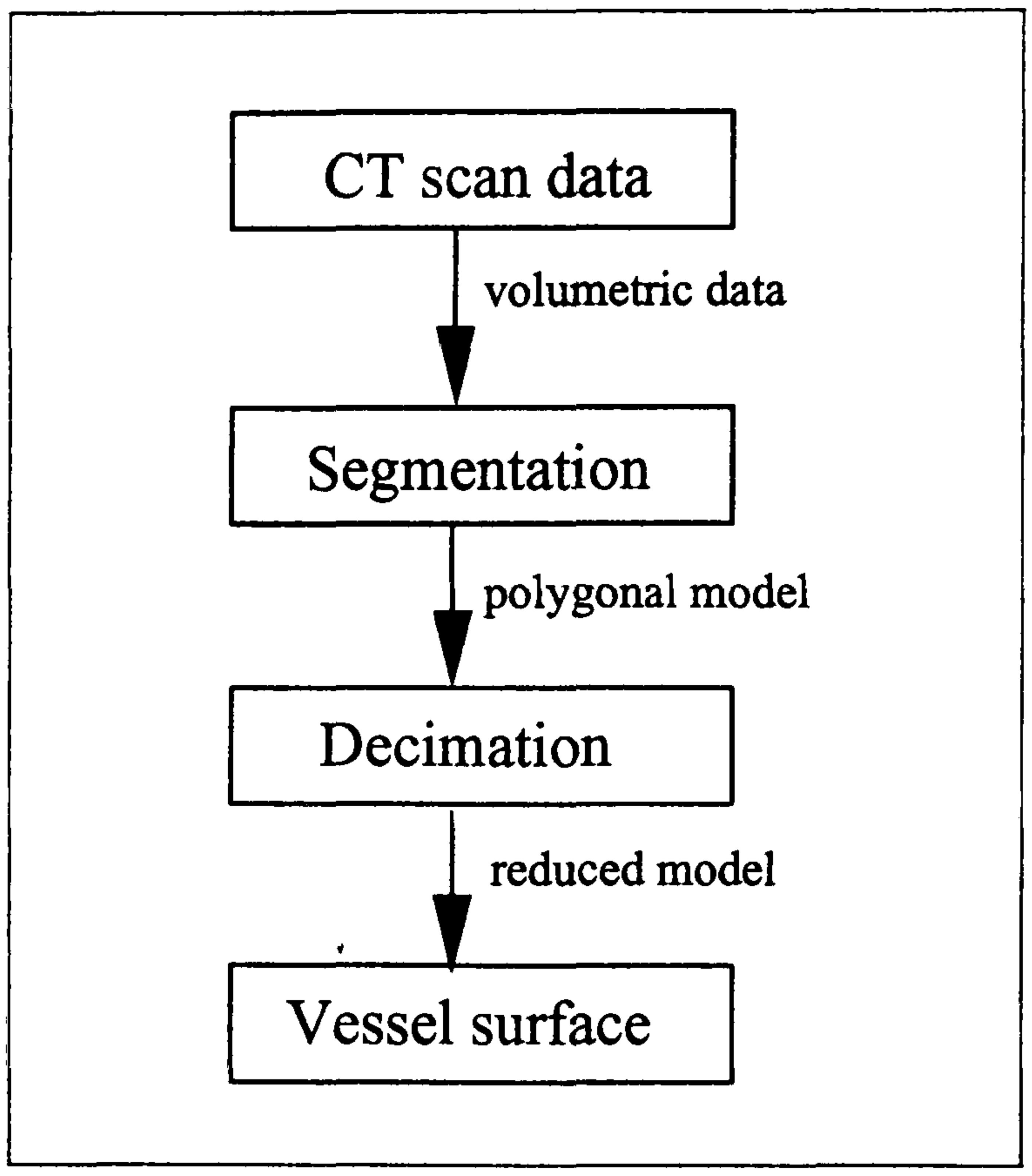


Figure 6.6: Modelling the geometry of the vessel.

we will develop a formal representation of the surgical tools' behaviour.

6.4.2.1 Theory of elasticity

To be able to create a computerised model of the object of interest, certain knowledge of the underlying science governing the behaviour of this object is necessary. *Continuum mechanics* is the discipline concerned with the analysis of the effect and the prediction of the outcome of applying external loads to a body [181]. A branch of continuum mechanics, which deals with the study of bodies described as possessing elastic properties is known as the *theory of elasticity.* Materials are described as elastic if they restore their original shape and size when external loads are removed. There are four basic concepts in the theory of elasticity:

- Displacement.
- Strain.
- Stress.
- Equilibrium.

Under the continuum mechanics discipline, the body is perceived as occupying a continuous region idealised as a set of connected points in 3D space. When external loads are applied to the body, these points change position to reach a state of equilibrium. The change of the position of point P to point P* under external load introduces the concept of *displacement.* If point P is described by the vector **r** from the origin of a fixed coordinate system and **r*** describes point P*, then the displacement $\mathbf{u} = \mathbf{r}^* - \mathbf{r}$ is the quantity that we seek to determine by utilising the theory of elasticity. The displacement quantity consists of two components. One component is described as the "rigid component" or "homogeneous deformation", since all parts of the body experience the same displacement regardless of their position. The idealised rigid body- which assumes that distance between body points is constant- experiences this type of deformation only [162].

The second component is known as "pure deformation" or strain. *Strain* is the change of the relative positions of two points per unit distance. For instance, suppose two points P_1 and P_2 of body (Ω) occupy position **r** and **r** + **dr** in their undeformed or natural state. When external loads are applied, the points change their positions to P_1^* and P_2^* described by the vectors $\mathbf{r}^*$ and $\mathbf{r}^* + \mathbf{dr}^*$ respectively- see Figure (6.7). In this case the strain E_{jk} is defined as follows:

$$(ds^*)^2 - (ds)^2 \equiv 2E_{jk}dx_jdx_k \quad where \ (j,k = 1,2,3) \tag{6.1}$$

where $ds = \mid \mathbf{dr} \mid$ and $ds^* = \mid \mathbf{dr}^* \mid$, which are the distances between points P_1 and P_2 and points P_1^* and P_2^* respectively; x_i are the scalar components of **r** [181].

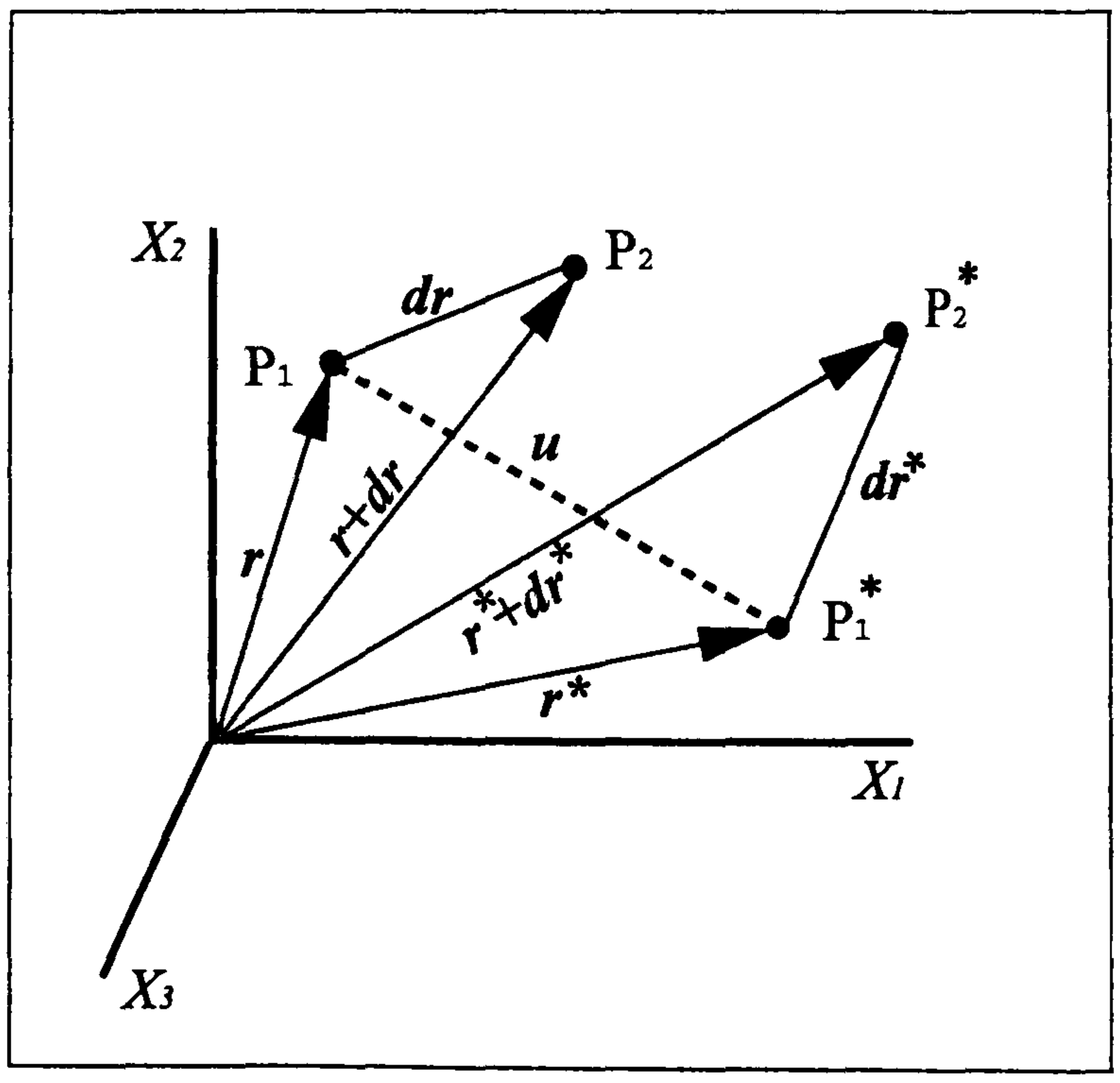


Figure 6.7: Strain representation.

In case of infinitesimal strain, E_{jk} can be approximated to e_{jk}, which is defined in terms of the displacement as :

$$e_{jk} = \frac{1}{2}(\frac{\partial \mathbf{u}_k}{\partial x_j} + \frac{\partial \mathbf{u}_j}{\partial x_k}) \quad where \quad (j,k = 1,2,3) \tag{6.2}$$

e_{jk} is known as the strain tensor, which describes the deformation of the body relative to the three coordinate axes- normal strain- in addition to shear strain. Equation (6.2) is the first basic relationship in the theory of elasticity- *displacement-strain relationship.*

When external loads cause deformation of the body, internal forces and moments develop in the body to resist the change in shape- *stress.* These forces can be divided into surface forces or body forces. Surface forces could be caused by pressure forces acting on the body and the *surface stress vector* is defined as

$$\sigma = \lim_{\triangle S \to 0} \frac{\triangle \mathbf{F^s}}{\triangle S} \tag{6.3}$$

where $\mathbf{F^s}$ is the applied surface force and $\triangle S$ is the surface area of a volume element.

Meanwhile, body force arises from the gravitational influence on the mass of the body. The *body intensity vector* is defined as:

$$\mathbf{f} = \lim_{\triangle V \to 0} \frac{\triangle \mathbf{F^b}}{\triangle V} \tag{6.4}$$

where $\mathbf{F^b}$ is the external body force and $\triangle V$ is the volume element.

According to Newton's third law, these forces must vanish when considering the whole body, yielding the second basic relationship in the theory of elasticity- the *equilibrium equation*

$$\int\int_S \sigma dS + \int\int\int_V f dV = 0 \tag{6.5}$$

Now, we can give a more accurate definition of an elastic body. If the elastic body is not influenced by external load, the body is free of internal stress and is in a natural state. Under the influence of external loads- with a certain loading limit- the state of the stress at a point of the body depends on the strain at that point. Finally, when the load is removed the body returns to the unstressed- or natural- state. This definition of an elastic body gives us the last basic relationship in the elasticity theory- the *stress- strain relationship*. This relationship is a generalisation of Hooke's law- which states that "the extension of spring-like bodies produced by tensile forces are proportional to the forces" [162]. In case of linear elasticity, this relationship is stated as:

$$\sigma_{ij} = C_{ijkl} e_{kl} \tag{6.6}$$

where C_{ijkl} is the elastic modulus constant, which is material dependent. This modulus has 81 elements, but it can be reduced to two constant values by utilising some mathematical features of stress and strain and assuming certain properties of the material [181]. The two constants- called *Lame' parameters*: λ and μ- can be found using the *Young's modulus (E)* and *Poisson's ratio* (ν) quantities, which are measured experimentally for different materials [71].

Combining equation (6.2), (6.5) and (6.6), we obtain a partial differential equation of the displacement **u** that describes the state of equilibrium in the body under some external load (**F**):

$$(\mu + \lambda)\frac{\partial^2 u_j}{\partial x_j \partial x_i} + \mu \frac{\partial^2 u_i}{\partial x_j^2} + F_i = 0 \quad where(i, j = 1, 2, 3) \tag{6.7}$$

Hamiltonian system

An alternative method of obtaining this equation is using *Hamilton's principle*, which states that: "the evolution of a dynamical system from time t1 to t2 is always such that the integral of the difference between kinetic and potential energy of the system is stationary over that interval". In case of elastic objects there exists a potential energy due to deformation, in addition to the potential energy due to external loads. Hence, the equation representing Hamilton's principle is :

$$\int_{t1}^{t2} U_\varepsilon - K - A_\varepsilon \ dt = 0 \tag{6.8}$$

where K is the kinetic energy, A_ε is the potential energy due to external forces and U_ε is the potential energy due to strain.

Once the representation of these energies are obtained for the dynamical system, the energy function (6.8) is minimised to solve for the displacement (**u**).

From the previous relationship we can deduce the *principle of virtual work* which states that: "the equilibrium of the body requires the equivalence of the total internal and external virtual work done by any imposed small virtual displacement satisfying the boundary conditions" [147]. The mathematical representation of this principle is as follows:

$$\int_V f\delta u dV + \int_{S\sigma} T.\delta u dS = \int_V \sigma_{ij}\delta e_{ij}\delta e_{ij} dV \tag{6.9}$$

Where $T = -\frac{\partial \mathbf{F}^s}{\partial u}$ and $f = -\frac{\partial \mathbf{F}^b}{\partial u}$; S_σ is the part of the surface where the load acts and δu is the displacement du. The first term represents the virtual work due to body forces. The second is the virtual work due to surface forces. While the right hand side is the internal virtual work due to stress.

Lagrangian mechanics

The Lagrangian mechanics utilise a *generalised coordinate* system for representing and solving the problem. This is an advantage in some cases where the use of cartesian coordinates is inefficient. The generalised coordinates could be cylindrical or spherical coordinates or any other independent variables of the problem. Once the generalised coordinates are specified, the equation of motion is transformed into these coordinates using the calculus of variations. Hence, the first step to obtain a Lagrangian equation of motion is to represent the cartesian coordinates (x,y,z) in the generalised coordinates q_i (where i=1,2,..n). Then, we represent the kinetic and potential energy of the system in the generalised coordinates. Finally, we use Hamilton's principle to get the equation of motion of the system in the new coordinate system:

$$\frac{d}{dt}\frac{\partial K}{\partial \dot{q}_i} - \frac{\partial}{\partial q_i}[K - V] = 0 \quad where\,(i = 1, 2, ..n) \tag{6.10}$$

where K and V are the kinetic and potential energy represented in the generalised coordinates.

6.4.2.2 Representing surgical tools

To model the behaviour of the surgical tool, it can be represented as a rod with constant cross section. To obtain the equation of motion, we analyse the distribution of stress over the rod when subjected to longitudinal forces- typically pushing and pulling the surgical tool- as well as torsion. In this case, we can simplify the distribution of stress as uniform over the cross section and ignore the influence of lateral contraction of the rod. This assumption is based on Saint- Venant principle [162], which states that:

If some distribution of forces acting on a portion of the surface of a body is replaced by a different distribution of forces acting on the same portion of the body, then the effect of the different distributions on the parts of the body sufficiently far removed from the region of application of the forces are essentially the same, provided that the two distributions have similar net force and moment.

Consequently, when the length of a rod is large in comparison to its cross sectional area, then the stress distribution over the ends has no appreciable influence on the characteristics of the solution in a portion of the rod far from the ends. In other words, because the ratio of the surgical tool diameter to its length is small, the distribution of the force acting on the tool at the insertion point has negligible effect on the tip of the tool.

Taking this assumption into account and deducing the equation of motion for the rod, Nowacki showed that the resulting equation is identical to the equation of a string subjected to longitudinal force [128]. Hence, we can use the set of equations of motion- presented in [128] chapter 3- for the string case to model the behaviour of the surgical tools. However, utilising these equations will create a specific model which can be used only in case of rods or string like objects. This contradicts the reusibility feature which we emphasised earlier. In order to preserve this characteristic in our model, it is recommended to use the general equation of motion- Hamiltonian and Lagrangian equations- in the representation stage and keep the model's specific details in the implementation stage.

In conclusion, the surgical tools are formally represented as one dimensional objects for the purpose of modelling their behaviour. A general equation of motion and deformation for elastic objects will be utilised to predict their behaviour. The physically-based modelling (PBM) field is a branch of computer science, which utilises laws of physics to model the behaviour of the object. Thus, we would like to utilise PBM techniques in our implementation stage. In the next section we will take a closer look at this field to explore the type of mathematical models utilised in its techniques and their applicability to our problem.

6.5 Physically-based modelling

Physically-based modelling (PBM) has been recognised as a new discipline in Computer Graphics in ACM SIGGRAPH'87 [35]. In PBM the physical properties of the modelled object are incorporated in the model, which will allow for the numerical simulation of the object behaviour. Modelled objects are affected by forces- in a force based system- or by energy- in an energy optimization system. In both cases, the object will behave according to physics laws and to their properties.

6.5.1 Physically-based modelling: Components

Despite the variety of PBM techniques, they all share basic components which reflect the basic concepts involved in continuum mechanics- described earlier. Thus, the concept of the object displacement is reflected in the *object representation* component. The concept of stress and strain is reflected in the *internal energy* component. Furthermore, the boundary conditions in the mechanical problem is equivalent to the *constraint* in the physically-based model. Finally, the equation of motion in the mechanical problem corresponds to the *mechanical formulation* component in the physically-based model. In both a physically-based model and a mechanical problem, numerical techniques are utilised to solve the problem. Figure (6.8) shows the components of the PBM approach. In the next few paragraphs we will explore these components in more detail.

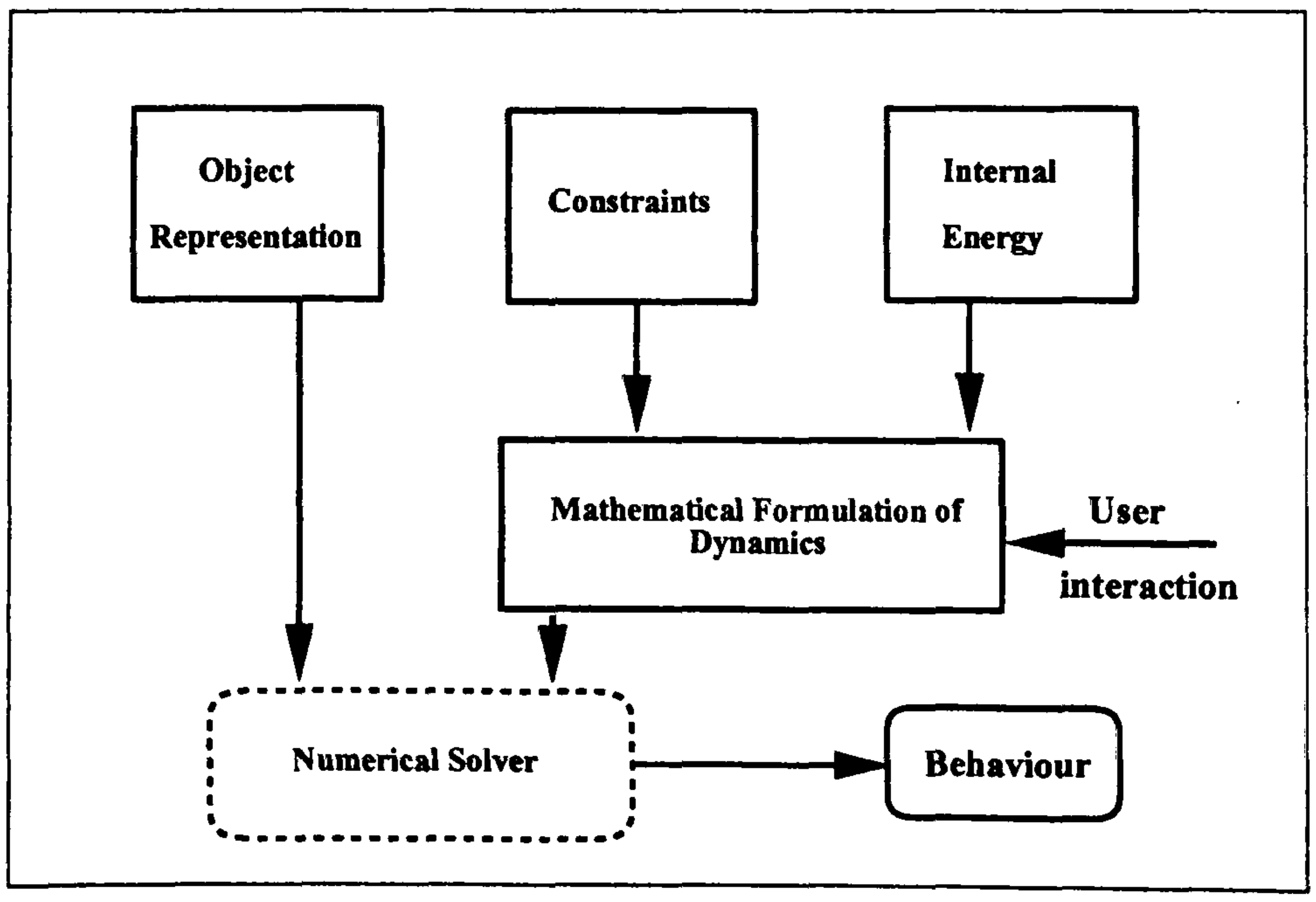


Figure 6.8: Components of a physically-based model.

Object representation

In this part of the physically-based model, the geometric properties- i.e. orientation, shape and position- of the modelled object are described. This geometry may be different from the actual shape of the object- e.g. in our case we have a 3D object, but we decided to approximate it by a 1D model. The functionality of this component is to provide the points of the model and their relative positions such that the distribution of the displacement in the model can be determined.

In general, the modelled object can be represented discretely as a set of particles- in a particle based system [190]. Conversely, in a continuum based representation, the object is represented in some continuous form- such as Bspline, NURBS or Bezier curve and surface [75]. Whether one chooses a discretised or a continuous representation, when the physically-based model is solved numerically, discretisation is necessary. The object representation has an effect on the accuracy of the model and on the performance of the system. For example, representing a cube made of spongy material as a set of six facets is inadequate since it cannot represent the distortion when the cube is squashed. On the other hand, some object representations are very elaborate such that computing numerical solutions for the model is expensive- as will be shown in the finite element method later.

Constraints

Constraints are used to control the behaviour of the model. Dynamics move the object under the influence of gravitational force downwards, however a constraint is what makes the object stop when it hits the ground. Constraints are forces introduced into the modelled system to enforce a certain behaviour or to prevent an unwanted one. Constraints can be dynamic, energy-based or reaction, depending on the method used to calculate the constraint.

Dynamic constraints generate forces using inverse dynamics. For example suppose the current acceleration of a system violates a constraint, the required forces are calculated to correct the acceleration of the system to a valid one. Barzel introduced a number of basic dynamic constraints to assemble and simulate the motion of a set of primitive rigid bodies [36]. However, dynamic constraints do not efficiently work with flexible objects, because of the complexity of the dynamics governing this type of object [114, 137].

Similarly, energy constraints are forces added to the system. The constraints are formulated as energy functions- as in the Hamiltonian equation (6.8); however, they do not necessarily represent the energy of the system. These functions are minimized using optimization theory to find a minimum which satisfies the constraint. For instance, if we would like to add a constraint to a system such that the volume of the object remains constant, we formulate a constraint function

$$g = V_{t_2}^2 - V_{t_1}^2 = 0$$

where V_{t_i} is the volume of the model at time t_i. In order to satisfy this constraint, we minimize the function (g) [137].

There are two types of energy constraint used in PBM. A penalty method is a technique that adds energy to the system to penalise the violation of a constraint. In fact, it introduces quadratic energies to the system, and this form of energy causes numerical instability [184]. The penalty constraint is triggered by the distance of the system from the constraint area. Consequently, the constraint is strongly applied when the system is far from its valid state, but when it gets closer to it, the constraint is very weak. Thus, the penalty method is not an exact constraint. Witkin and et.al. [185] presented a similar set to Barzel's dynamic constraint but using the penalty method. They recognised the drawbacks of this method; however, it does have the advantage that it can support multiple constraints without having to worry about over-constrained or under-constrained systems- i.e. systems with contradicting constraints that makes them unachievable or systems with too few constraints such that their behaviour is incorrect.

The second form of energy constraint is an augmented Lagrange constraint (ALC). This constraint is a hybrid between the penalty method and Lagrange multipliers (LM), which enhances the convergence capabilities of the LM method. Constraints in ALC are formed into a differential equation that is added to the system equations. ALC is more suitable when trying to control the properties of the material modelled- for example its elastic or mouldable behaviour [137].

Reaction constraints have superseded energy constraints, because unlike the penalty method, this method fulfills the constraints fast and exactly, and unlike ALC it does not require the addition of equations to the system. Reaction constraints will generate forces that cancel the violating force and replace them with constraint forces. Hence, it is simple and reliable, but it is limited. Reaction constraints can only apply one constraint to one point of the model. Nevertheless, the approach has satisfactory performance when imposing non-penetration constraints on an object [114, 137].

Internal energy

The internal energy component deals with stress and strain initiated in the model due to deformation. In elastic models, the internal energy is what causes the shape of the model to return to the natural shape after the forces are removed. In general, studying the distribution of stress and strain in the model is the way most engineering models calculate the internal energy in the model [61]. Other ways of calculating the internal energy in the model include using differential geometry of the object representation [174] and Hooke's law [78, 124].

Mechanical formulation

In the physically-based model, the behaviour of the object is calculated based on a physical law. This part of the model includes the mathematical representation of the equation of motion adopted by the model. As shown before this can be represented using the Lagrange mechanics equation (6.10) or the Hamiltonian equation (6.8) or simply Newton's Law:

$$\mathbf{F} = m.\mathbf{a} \tag{6.11}$$

where m is the mass of the object, $\mathbf{a}$ is the acceleration and $\mathbf{F}$ is the net force acting on the object.

No matter what formulation one chooses, the behaviour of the model is the same. However, the numerical method used to solve these equations is different and that can affect the efficiency of finding a solution [75].

6.5.2 Physically-based modelling: Techniques

Physically-based modelling techniques are numerous, and we will not attempt to present a full survey of all these methods. However, we will review the most popular techniques.

6.5.2.1 Mass-spring method

The mass- spring method is an example of a particle-based system. The modelled object is represented as a collection of points that have mass, position and velocity.

The Newton equation can be used to represent the motion of the system:

$$m.\mathbf{a} = -d.\mathbf{v} - k.\mathbf{r} + \mathbf{F} \tag{6.12}$$

The left hand side of the equation represents the inertial force of the mass point, while the right hand side represents internal and external forces ($\mathbf{F}$) affecting the system. The internal energy is represented using Hooke's law by including massless springs between the mass points- see Figure (6.9). Springs are used to restore the shape of the model after deformation. The spring will deform in proportion to the applied force affecting the mass point. The spring modulus k quantifies the elasticity of the spring. Usually, a dashpot effect- damping- is added to the system to eliminate the oscillation of the spring and to reduce the drifting of the numerical solution. Damping is proportional to the velocity of the mass and d is the damping coefficient value [114].

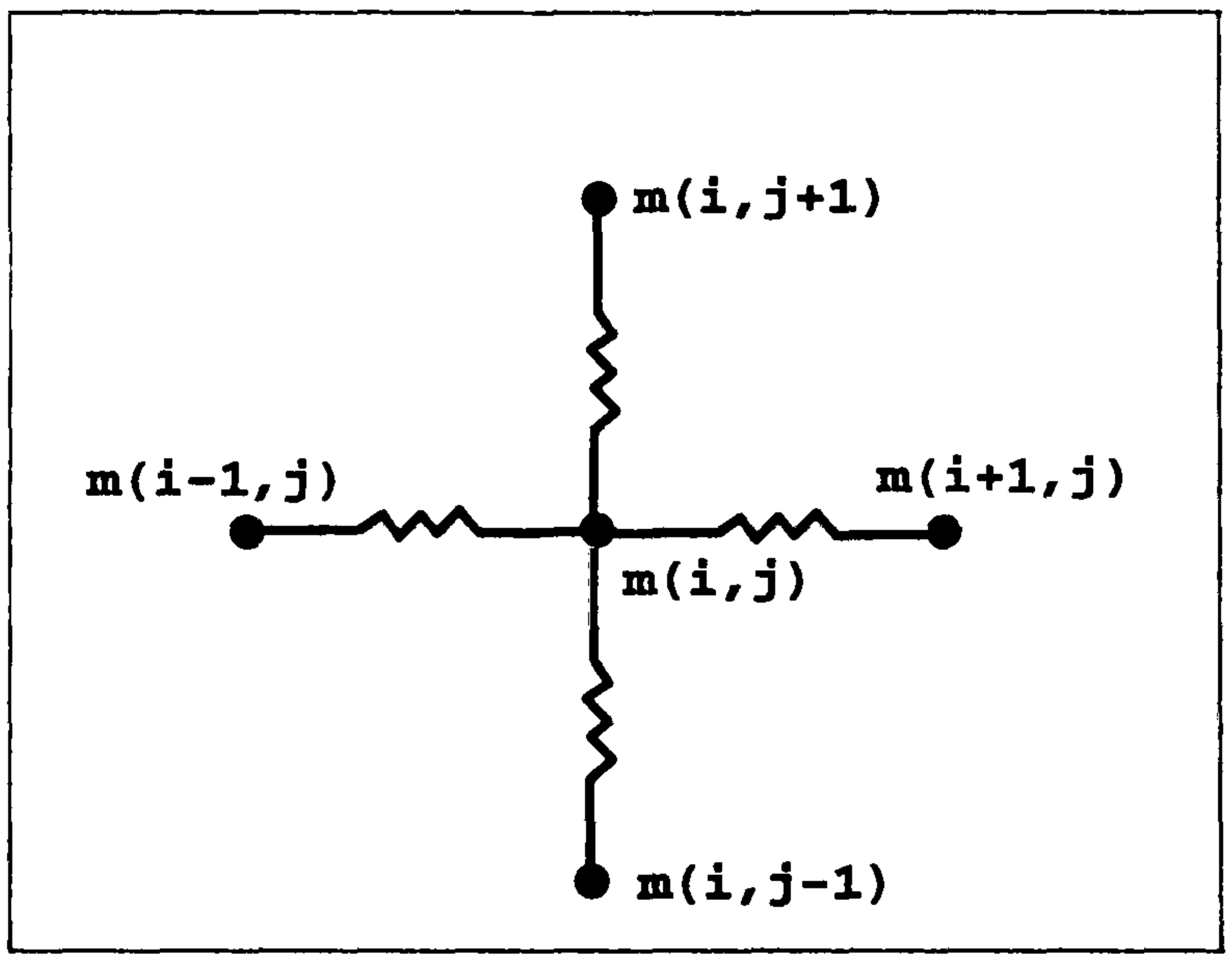


Figure 6.9: Mass- spring model.

This method is easy to implement and fast to compute, therefore, it is widely used in different applications- e.g. [122, 78]. Nevertheless, this modelling method possesses some drawbacks because of its simplicity. The major disadvantage is its numerical instability. This instability is most evident when the rigidity of the modelled object is high. To simulate the behaviour of such object, the spring modulus must be increased, which decreases the stability of the system [115]. The stability of the system is also decreased when more mass points are added to the model. This is due to the fact that the spring system is an energy minimisation technique but on the local level of a mass point. Hence, every mass point in the system will try to reach equilibrium; when the number of points are increased, more error is accumulated in the system leading the model to incorrect behaviours.

Apart from the stability of this method, it is not intuitive to use the spring method in representing some elastic objects. This is due to the lack of correspondence between the spring modulus and the natural way of representing elasticity properties of an object such as bending, torsion and stretching coefficients. This means that trial and error is required to obtain the correct k and d values to quantify the desired behaviour.

Miyazaki proposed another draw back of the spring model; in response to the large collision forces, springs may experience a change in their arrangement- i.e. mass points connecting the springs. He proposed to solve this problem by introducing a factor of rigidity to the spring model when handling collision [124].

6.5.2.2 Finite element method

The numerical finite element method (FEM) has been widely used in the physically-based modelling field because of its well known numerical stability. Using this technique allows the use of a continuous object representation- e.g. parameterised implicit function [65], B-spline curve [75]- and enables the physical properties to be described continuously over the object. However, to determine the motion or the deformation of the objects, the FEM discretises the object into a number of basic elements- e.g. cube, prism or tetrahedron [114]- see Figure (6.10). Each element consists of a set of nodes connected by edges. To propagate the effect of a force applied on one node to the neighbourhood, a stiffness matrix K is composed from the aggregation of individual elements' matrices. This element decomposition has given the FEM its strengths and weaknesses. Because FEM uses a global stiffness matrix to determine the object shape and motion, it actually finds the global minimum of the energy equation- unlike the mass-spring method- which makes it accurate. On the other hand, the huge size of these matrices makes FEM inefficient in terms of storage space. Furthermore, the cost of inverting these matrices makes FEM computationally expensive. Another disadvantage of the finite element structure is that collision is of O(mn) where n is the number of elements and m is the number of faces. However, both Essa and Pentland suggested using volumetric collision detection- as will be described later- to solve this problem [65, 136].

The FEM utilises the Hamiltonian system with the principle of virtual work mentioned earlier to predict the behaviour of the object. Hence in the static case the equation of deformation is represented as:

$$K.\mathbf{u} = \mathbf{F} \tag{6.13}$$

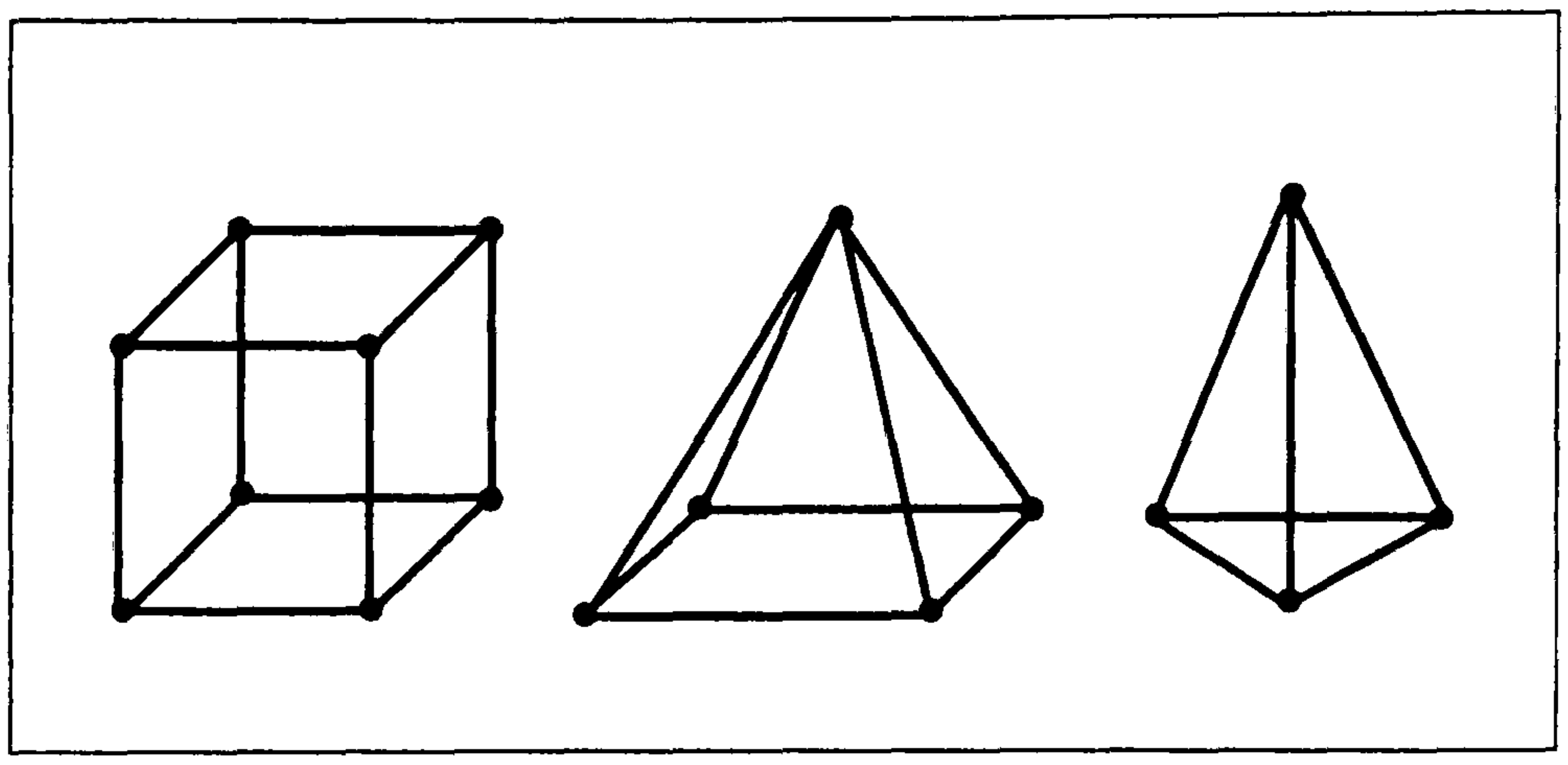


Figure 6.10: Basic elements in a FEM.

where $\mathbf{F}$ is a vector of the loads acting on each element in the model, $\mathbf{u}$ is the unknown displacements of the element nodes and K is the stiffness matrix which holds the physical properties of the object [114, 74].

On the other hand, when representing a dynamic system, the inertia of the object, acceleration and velocity are taken into account in the energy function:

$$M.\ddot{\mathbf{u}} + D.\dot{\mathbf{u}} + K.\mathbf{u} = \mathbf{F} \tag{6.14}$$

where M and D are the mass and damping matrices for the elements of the model.

The above two formulations suffer from a great disadvantage: the computational cost of calculating the solution. Pentland and Chapman suggested the use of a modal analysis technique to reduce the cost of computation. Using this method implies the reformation of equation (6.14) in terms of the eigenvector and eigenvalue as follows:

$$\lambda\phi = M^{-1}K\phi \tag{6.15}$$

where λ is the eigenvalue and ϕ is the eigenvector [136].

The model is decomposed into a set of vibrational modes and the response of the object to the applied forces is a linear composition of these modes. This method can save CPU time when calculating the behaviour by reducing the number of modes and frequencies used to represent the object- as is shown by Chapman in [46, 47]. However, it does not work for complex shapes or complex loads. Furthermore, it does not solve the storage problem that FEM suffers from.

6.5.2.3 Implicit functions

An implicit surface is defined as a set of points $\mathbf{p}$ such that $f(\mathbf{p}) = 0$ where $\mathbf{p}$ is defined in 3D space. The function f is called an implicit function. Implicit functions have been used for a long time to represent the geometry of an object. They are favoured- from the geometric point of view- for their efficient storage of complex objects. The implicit function f divides space into inside and outside the surface, such that $f(p) < 0$ is the function which defines the volume inside the surface- this function is known as the inside/outside function. Implicit representation has proved useful in physically-based modelling, especially in collision detection and response. This is due to the fact that the inside/outside function provides an accurate and fast method of checking inter-penetration; while this function together with the stress-stain relationship described earlier in section (6.4.2.1) can be used to compute the response forces due to penetration [65].

Essa et.al. [65] used implicit function representation of superquadrics surfaces. Meanwhile, implicit surfaces produced by the meta-ball method have been used in [54]. This method models the object as a set of seeds (or atoms), which possess a specific distance function- called the field function. The atoms can be any shape- e.g. curve, polygons, etc [114]. The field function characterises the seeds physically and geometrically; radius, stiffness and thickness are used for this purpose in [54]. Figure (6.11) shows an implicit surface generated by two seeds and the field function used to describe the seed properties, where R is the radius and k is the stiffness.

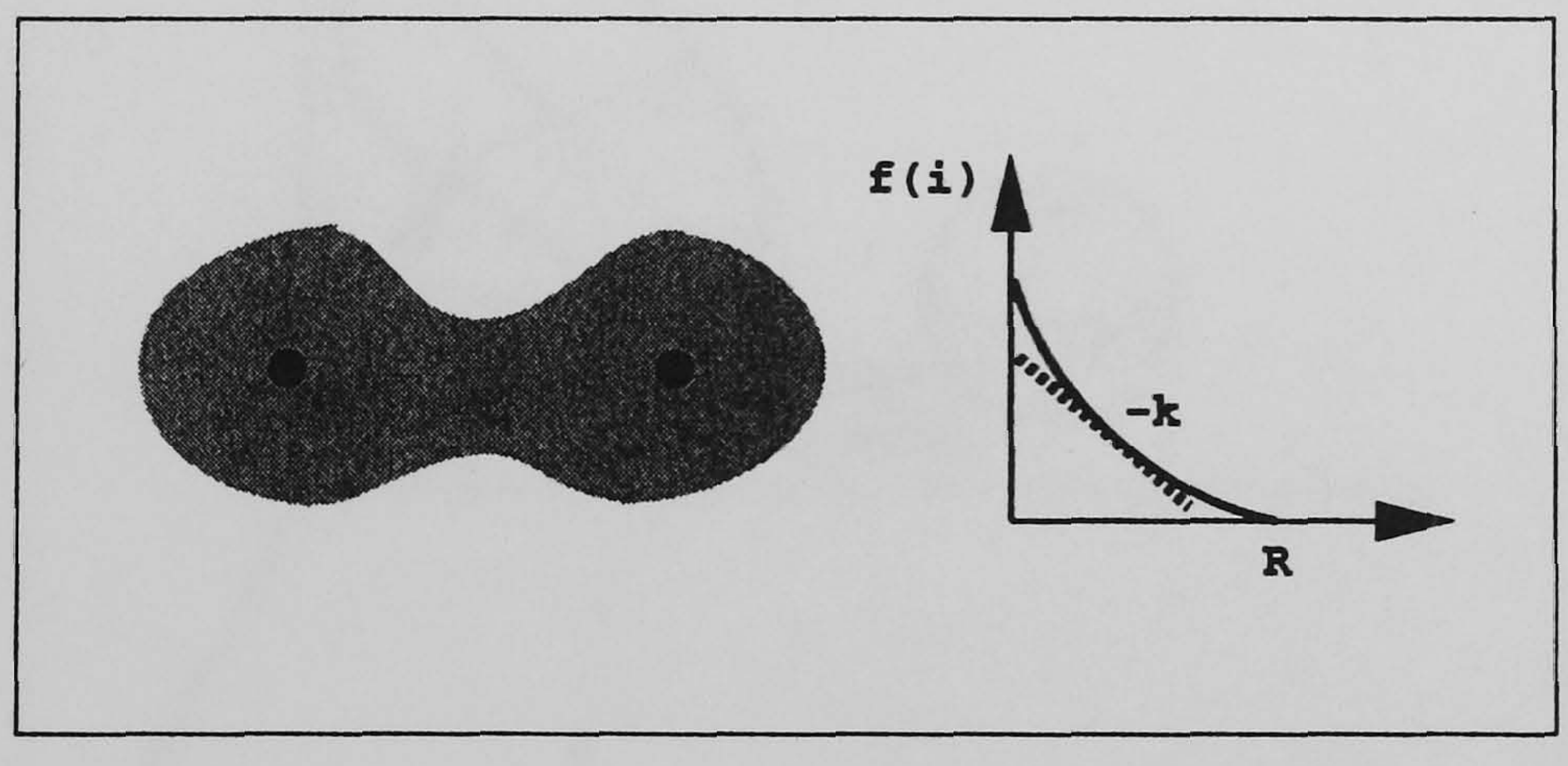


Figure 6.11: Implicit surface model.

Implicit models can be deformed using FEM [65] or using global matrix deformation [186, 30]. On the other hand, [54] employs a different method of calculating the model deformation. In this work, implicit surfaces are used as an elastic coat for any particle system. The seeds are treated as rigid objects and their motion is computed with the appropriate equations. The change of the seeds' positions will result in the elastic deformation of the implicit surface.

As mentioned before, implicit surfaces are efficient in handling collision of flexible objects. However, because only global deformation can be applied to them, the model can exhibit limited deformation.

6.5.2.4 Primal and hybrid methods

Terzopoulos has proposed several techniques to model the motion and deformation of non-rigid objects, for the purpose of animation and computer vision. One technique- the *primal model-* is suitable for simulating non-linear elastic materials- e.g. cloth and rubber [174]. The object modelled using this method is represented as a set of points in 3D space. A point P can be described using the material coordinate, which depends on the dimensionality of the object. For example, in the case of the two dimensional surface- shown in Figure (6.12)- P has two material coordinates (a1,a2). On the other hand, P is described in terms of fixed coordinate system using the vector **r**. Points of the model represent concentration of mass in the model. The displacement of these points raises stress (internal energy)- as described in the mechanical problem in section (6.4.2.1). Figure (6.12) shows the model in a rest state- where P is described by the vector $\mathbf{r}^0$, then after deformation- P^* is described by the vector **r**.

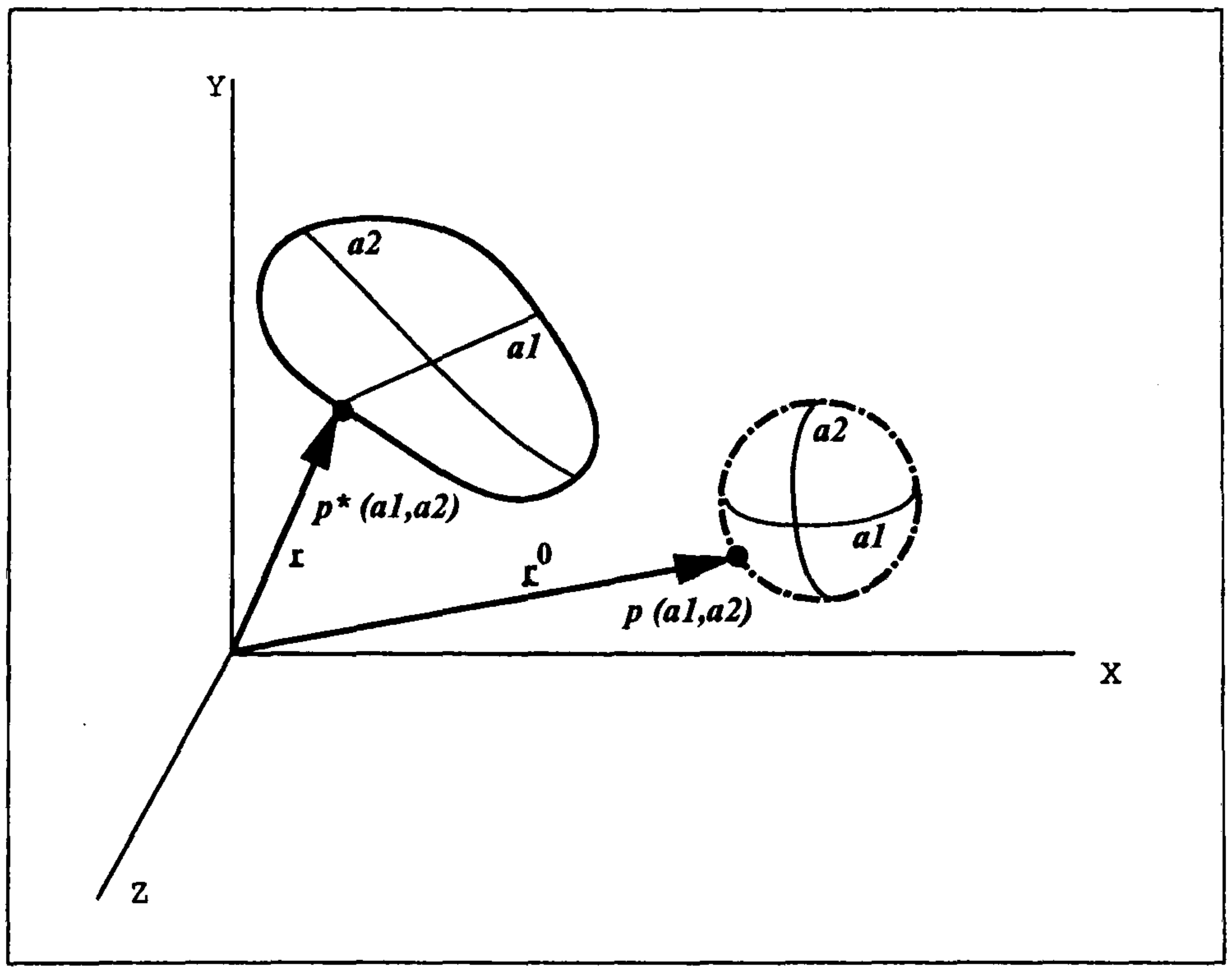


Figure 6.12: Terzopoulos's primal model.

This technique uses the Lagrange formulation to represent the equation of motion.

$$\frac{\partial \mathbf{r}}{\partial t}(m\frac{\partial \mathbf{r}}{\partial t}) + \gamma\frac{\partial \mathbf{r}}{\partial t} + \frac{\delta\xi(\mathbf{r})}{\delta \mathbf{r}} = \mathbf{F} \tag{6.16}$$

where m is the mass of the point, γ is the damping of the velocity of the mass point. **F** is the applied load. The new position **r** of the mass point is the unknown variable. The term $\delta\xi(u)$ is the variational derivative of the internal energy which emerges in the model due to strain. This energy is calculated using concepts of differential geometry.

Classical differential geometry is concerned with studying the local properties of shapes- i.e. the properties which depend on the behaviour of the shape in the neighbourhood of some point. For example, in differential geometry a curve in 3D space described in terms of its arc length s is fully characterised by the arc length, the curvature $\kappa(s)$ and the torsion $\tau(s)$. Thus two curves possessing the same values of these parameters are different only in their rigid motion. Similarly, a surface is parameterised by two tensors- called the metric and curvature tensors- while a three dimensional solid is parameterised by a metric tensor [55]. Calculating the internal energy based on these parameters makes the energy invariant to the rigid motion of the object [174].

As mentioned before, the primal technique has been successful in modelling highly elastic behaviour. However, the numerical stability of this method degrades when the rigidity of the object increases. For this case, Terzopoulos has proposed the *hybrid model* [175]. In this method, points in the object are represented as a reference component $\mathbf{r}$ and an elastic component $\mathbf{e}$. The object frame (xyz) as well as the reference component changes position and orientation under the influence of the rigid motion of the body. Meanwhile, deformation in the object shape is represented as a displacement from this reference component- see Figure (6.13). The local position of P with respect to the object frame is represented by $\mathbf{q}$, while $\mathbf{x}$ represents the global position of P with respect to the fixed world frame (XYZ).

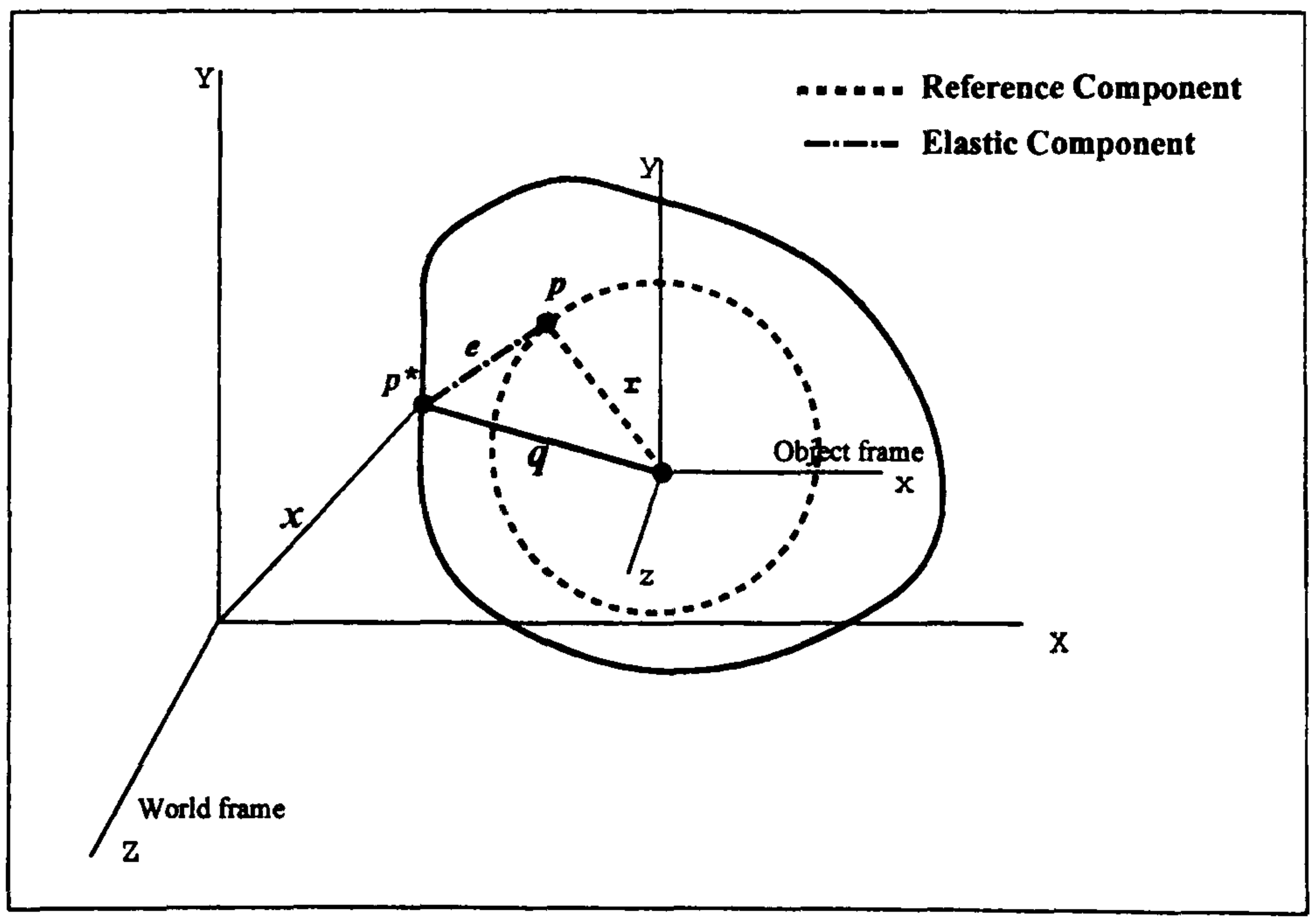


Figure 6.13: Terzopoulos's hybrid model.

Unlike, the primal method, the inertial energy used in this method is linear and is calculated from "controlled continuity spline" energy [175]:

$$E = \frac{1}{2}\Sigma_{m=0}^{p}\Sigma_{|j|=m}\frac{m!}{j_1!...j_d!}w_j(u)|\partial_j^m e|^2 \quad where|j| = j_1 + ... + j_d \tag{6.17}$$

This formulation produces a weighted function of the partial derivative of the elastic component **e**, such that p controls the order of the derivatives used and d represents the dimensionality of the model. Again, the weights of the energy function reflect material properties- e.g. curvature, stretching. The advantage of this technique is the coupling between the rigid motion and the elastic deformation. The following system of equations, are used to solve the motion under this method. The first two equations govern the rigid motion of the body, while the last equation represents the elastic deformation of the body.

$$m\dot{\mathbf{v}} = \Sigma\mathbf{f} - \frac{d}{dt}\Sigma\mu\dot{\mathbf{e}} - \Sigma\gamma\dot{\mathbf{x}} \tag{6.18}$$

$$\frac{d}{dt}(I\mathbf{w}) = \Sigma\mathbf{q}\times\mathbf{f} - \frac{d}{dt}\Sigma\mu\mathbf{q}\times\dot{\mathbf{e}} - \Sigma\gamma\mathbf{q}\times\dot{\mathbf{x}} \tag{6.19}$$

$$M.\ddot{\mathbf{e}} + D.\dot{\mathbf{e}} + K.\mathbf{e} = \mathbf{f} - \mu\dot{\mathbf{v}} - \mu\mathbf{w}\times(\mathbf{w}\times\mathbf{q}) - 2\mu\mathbf{w}\times\dot{\mathbf{e}} - \mu\dot{\mathbf{w}}\times\mathbf{q} \tag{6.20}$$

where :

- m is the total mass of the object as if it was concentrated in one point.
- μ is the mass of one of the points comprising the object.
- I is the inertia tensor of the object.
- **v** and **w** are the linear and angular velocity of the centre of mass of the object respectively.
- $\dot{\mathbf{x}}$ is the velocity of a mass point.
- **f** is the applied force on a mass point.
- **q** is the local position of a mass point.

- M, D and K are matrices of size NxN- where N is the number of mass points in the model- representing mass, damping and stiffness of the model respectively.

An extension of the hybrid method has been used to model inelastic behaviour. This behaviour has been achieved by modifying material properties and reference component during the deformation process [172]. Both the primal and hybrid methods- together with other extensions of the same concept- have been used in various applications to model object behaviour which ranges from totally rigid to totally elastic and plastic materials [170, 171, 76, 77].

6.5.3 Physically-based modelling: Applications

Since the introduction of the PBM techniques, the animation and vision fields have been the driving and directing force of the PBM field. PBM techniques have been utilised to create physically accurate motion of characters, animals and objects [177, 111, 122, 45]. The simulation of non-rigid objects like cloth is now possible with greater accuracy [176, 61, 179]. Perhaps the most evident example of success achieved in this area is the latest short film produced by Pixar in SIGGRAPH'98 - "Geri's Game".

Similarly, in vision, PBM techniques are popular in extracting information from images- segmentation- and shape recognition for the purpose of tracking or matching [119]. In fact, the segmentation method which we use for extracting the aorta from the CT slices- described in section (6.4.1)- is an example of the application of PBM techniques in vision. This technique and others [173, 59] utilise a *deformable model* in their technique. A deformable model is a 3D mesh of triangles, where the position of the vertices of the triangles comprise the unknown displacement. External forces and constraint forces cause the deformation of the model, which is computed using a mechanical formulation as any other physically-based model. What makes deformable models special is that external forces are produced from two dimensional images. For instance, if the purpose of the model is to track the surface of an object- represented in the 2D image- then the gradiant of the image is used to force the model towards the desired surface.

In the medical field PBM techniques are utilised to process medical data and to simulate structure behaviour and pathology. In fact most of the medical applications reviewed in section (2.2) have used a PBM technique for one purpose or another. Now that we have enough background about this field, we can report the techniques utilised in these applications.

The finite element method has been used in different simulators to model the deformation of non-rigid organs- e.g. eye corneal tissue in [93], liver in [51] and soft tissues in the knee [113]. In the endovascular surgical training- daVinci- the surgical tools were modelled as a non-linear FEM with beams as the basic elements of the model [2]. They also utilise the Lagrange equation with a spherical coordinate system to model the dynamics. Finally, the FEM has been used in [164, 163] to model tissue cutting in surgery simulators.

Cover used a physically-based model to simulate the behaviour of the gall bladder in a laparoscopic procedure. Springs are used for internal energy representation and an energy minimisation function is formulated to calculate the dynamics [52]. On the other hand, the generic surgical simulator- Teleos- presented by HT Medical utilises the mass-spring model to simulate the behaviour of the organs, which are represented using polygonal models.

Another utilisation of the PBM techniques is in the visualization system presented in [82] for simulating blood flow in the aneurysm. The blood is modelled as a 3D incompressible Newtonian object. Partial equations are used to determine the dynamics of the blood within the boundary conditions of the vessel.

In addition to utilising these techniques in simulating deformable objects and segmenting static structures, deformable models have been used to track dynamic structures. For instance, [22] tracked the motion of the left ventricle of the heart to diagnose heart diseases. He utilised curves and surfaces as basic elements of the deformable model and used differential geometry to calculate the internal energy in the model.

6.6 WebSTer's physically-based model: Design

Now that we have completed the abstraction and representation stages of our model and accumulated the necessary background information about the theory which governs the behaviour of the surgical tools and the techniques used to simulate such a behaviour, we are ready to desgin and implement our simulation engine which will reside on the server side.

In section (6.5.1), we mentioned that the PBM techniques are a composition of some object representation, constraint formulation and an internal energy formulation for non-rigid objects, all together with some mechanical formulation of dynamics. The different combination of these components- provided they are compatible- yields a specific technique, that could be more flexible, more accurate, faster or more suitable for certain types of objects.

Rigid object motion has been well established, since computer graphics has been dealing with rigid bodies since its inception. Unfortunately, rigid objects are not the only type of object around us. Consequently, in a virtual environment or in a simulator system, one may need to model both rigid and non-rigid objects. However, non-rigid objects can have a wide variety of behaviours- e.g. elastic, visco-elastic, mouldable, thermo-elastic... etc. Mostly, techniques to model these behaviours are as young as the PBM field. In addition, those techniques depend largely on trial and error to choose parameters that are used to quantify the model. The behaviour of the model also depends on the initial conditions of the system, which again is determined by trial and error. Furthermore, some techniques are more suitable than others to model certain behaviours. In fact there is always this trade off between accuracy and efficiency, as is the case in most computer graphics techniques. Hence, the modeller may feel obliged to attempt a few techniques before deciding what is best for their model.

From the previous paragraph, we can sense the need to model both rigid and non-rigid objects in a unified way that allows the experimentation with different modelling techniques. In this case, what is required is a unified design architecture that can be used to model rigid or non-rigid behaviours. In addition, the design should be modular, where different modules represent different components of the PBM approach. Equally important is to impose a unified communication strategy between the different modules, such that the different approach of some module can be used without affecting the structure of the other modules. In this section we present a generic design architecture that meets these requirements. The proposed design is a generalisation of the hybrid technique presented in [175] and described in section (6.5.2.4).

The proposed design architecture can be generically used to model rigid bodies and elastic bodies. This is achieved by dividing the dynamic process into two layers. On one layer, the rigid dynamics based on Newton's formulation ($\mathbf{f}= m\,\mathbf{a}$) is used to handle the motion of purely rigid bodies and the rigid component of the displacement exhibited by elastic objects. In this layer, the detailed object representation is of no interest to us. Thus, the total mass of the body and the inertial tensor is as much as we need to know about the modelled object. This layer affects the velocity and acceleration of the centre of mass of the object.

In the second layer, Lagrangian dynamics is imposed on the non-rigid object. The detailed representation of the elastic object is of interest to us in this layer;

however, we try to hide this representation from the other components to achieve the modularity requirement. The output of this layer is manifested in the deformation of the elastic object. Figure (6.14) shows the proposed design architecture.

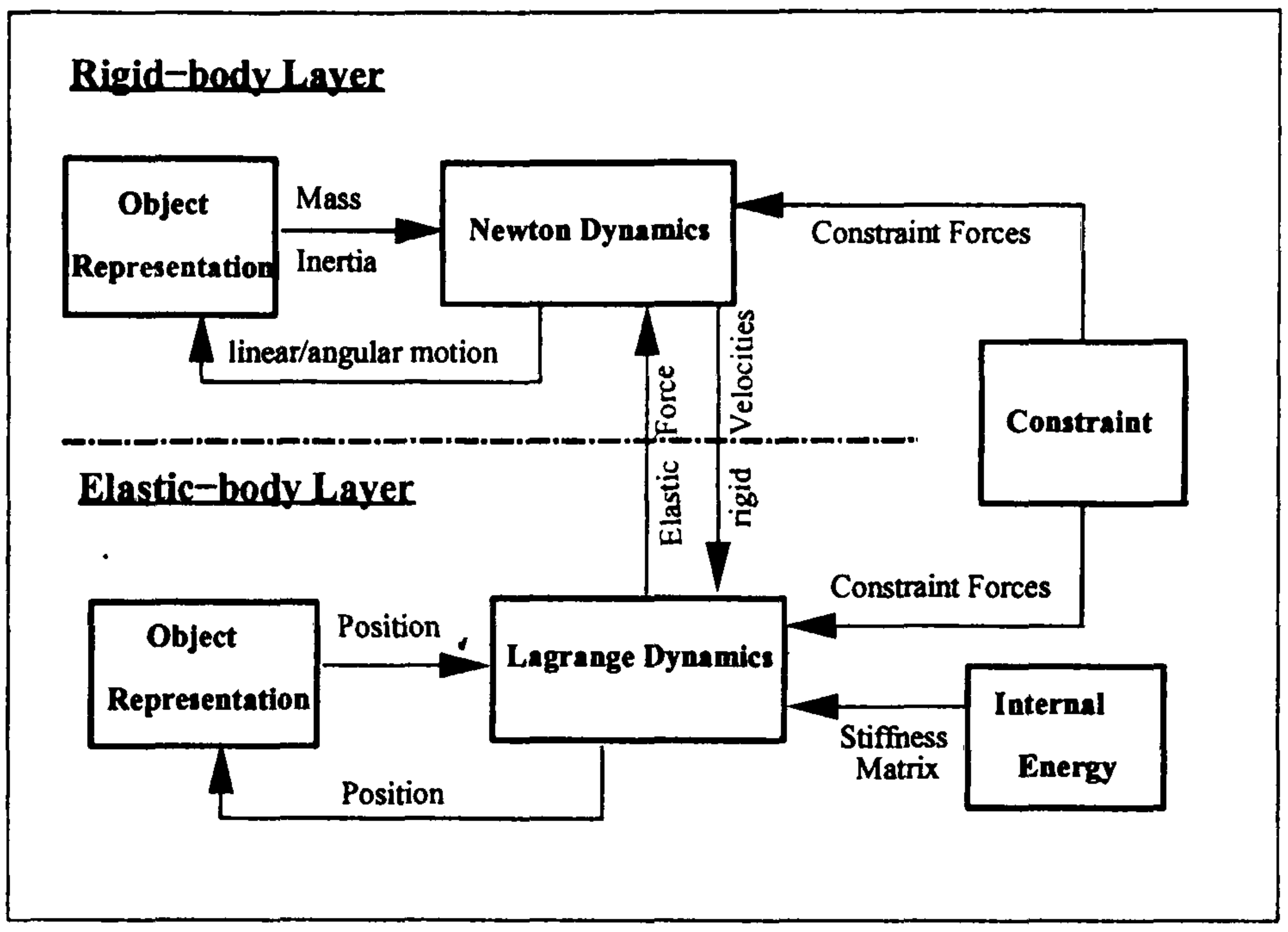


Figure 6.14: Generic PBM design architecture.

To facilitate the experimentation of different elastic methods, the elastic layer components are treated as black boxes. However, they are expected to represent their results in certain formats to allow communication between the components. The core of the elastic layer is the Lagrangian dynamic formulation of an object motion represented by the following equation.

$$M.\ddot{\mathrm{u}} + D.\dot{\mathrm{u}} + K.\mathrm{u} = \mathbf{F} \tag{6.21}$$

To solve this equation, each component is expected to contribute part of the information needed. For instance, the object representation module contributes a vector of unknown quantities that needs to be computed $\mathbf{u}$. The interpretation of this vector is not the responsibility of the dynamic module. Thus, the vector could represent the position of a set of mass points in a mass-spring system, or it could be the displacement of nodes in a FEM or points in a finite difference representation. However, it is the responsibility of the object representation module to interpret this vector, compose and update it as appropriate. In addition, the object representation module formulates the mass M and damping D matrices for the particular model.

In a similar fashion, the internal energy module is responsible for calculating the potential energy of the elastic model. Differential geometry, spring or strain energy could be used to achieve this goal. The output expected from this module is a stiffness matrix K.

Finally, the constraint module is expected to enforce the desired constraints on the system and to output correction forces to the dynamics module. To specify that the output of this module should be in force-based format is not a restriction on the methods that can be used to solve constraints. This is due to the fact that most methods- even energy based methods- can be formulated in such a way as to produce forces [137]. Although the internal energy of an elastic object can be considered as a constraint, we have separated it because of the different required format of this information.

The output of the dynamics module is the vector representation of the object, which reflects the deformation of the shape under the applied loads. Again, it is the responsibility of the object representation module to interpret this vector either by directly changing the position of the mass points or by changing the position of the control points, which consequently affects the shape of the object.

User applied forces and torque are handled in both the rigid and the elastic layer, because they have an effect on the object motion and deformation. Similarly, some constraints- such as collision response- have a dual effect on the object, which means that their forces are applied to both layers.

Elastic object motion and deformation are not two separate entities. In fact, each entity affects the other. This requires that both layers in our design exchange enough information to the other layer to achieve the correct behaviour. The elastic layer should pass a vector of elastic forces which contributes to the rigid dynamics. On the other hand, the velocity and acceleration of the centre of mass is passed to the elastic layer. When modelling a rigid object the elastic forces are set to zero.

6.7 WebSTer's physically-based model: Implementation

We have used the generic design described in the previous section to experiment with different PBM techniques to choose the most suitable technique for modelling our surgical tools. Mass-spring and primal approaches were implemented. Details of these implementations are presented in the next section. In section (6.7.2) we present the implementation details of the hybrid model which we adopted. Then in section (6.7.3), we will report the non-penetration constraint that we utilised in our simulation.

6.7.1 Experimentation methods

Mass-spring method

Under the mass-spring approach, the 1D tool is represented as a number of mass points. Spring structures are added between the mass points to spread the effect of applied forces around the neighbouring points. Figure (6.15) shows the tool representation under this model.

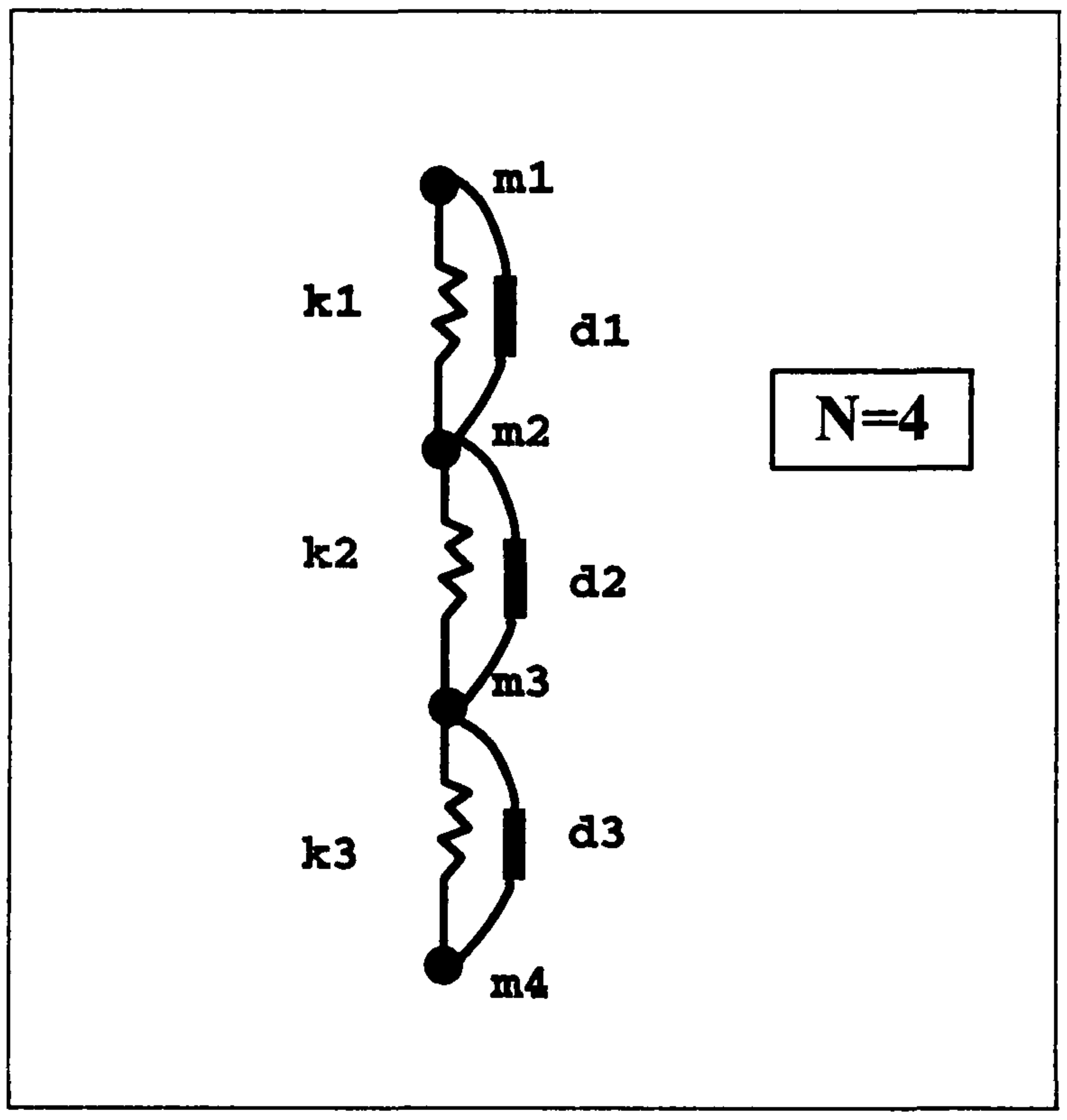


Figure 6.15: The mass-spring representation of the model.

The spring modulus k_i is used to construct the stiffness matrix of the model. The matrix is constructed such that middle mass points are affected by the tension/ compression of the two springs attached to it. End points are only affected by one spring. The stiffness matrix is shown below- for the model shown in Figure (6.15)- when springs are used to represent the internal energy.

$$\begin{bmatrix} k1 & -k1 & 0 & 0 \\ -k1 & k1+k2 & -k2 & 0 \\ 0 & -k2 & k2+k3 & -k3 \\ 0 & 0 & -k3 & k3 \end{bmatrix} \tag{6.22}$$

Dampers are added between mass points as well, such that the damping matrix looks as follows:

$$\begin{bmatrix} d1 & -d1 & 0 & 0 \\ -d1 & d1+d2 & -d2 & 0 \\ 0 & -d2 & d2+d3 & -d3 \\ 0 & 0 & -d3 & d3 \end{bmatrix} \tag{6.23}$$

Meanwhile, the mass matrix is a diagonal matrix where mass values of each point are the elements of the diagonal.

$$\begin{bmatrix} m1 & 0 & 0 & 0 \\ 0 & m2 & 0 & 0 \\ 0 & 0 & m3 & 0 \\ 0 & 0 & 0 & m4 \end{bmatrix} \tag{6.24}$$

We then implement equation (6.12) in the mechanical formulation component and use a numerical package to solve for the new position of the mass points. Because the surgical tools exhibit no significant stretching effect when longitudinal forces are applied to one end of it, we had to increase the spring modulus value. This produced unrealistic simulation of the surgical tools which discretely jumped rather than moved smoothly; in addition to occasional instability of the structure when the forces applied exceeded the spring stiffness- as reported in [124].

Primal model

Terzopoulos's primal model was also used to model the 1D tool. The model is discretised as a set of points which are described with respect to two coordinate systems: the local coordinates use one parameter a to refrence the points of the one-dimensional model, meanwhile the fixed world coordinates describe the points of the model as 3D position in space $\mathbf{r}$. The arrangement of the points at time=0 represents the rest shape of the model. When forces are applied to the model, such that the set of points are displaced, deformation energy builds up to counteract the forces and to restore the rest shape of the model. Internal energy is represented using differential geometry for curves using the following equation:

$$E = \int_{\Omega} \alpha(s - s^0) + \beta(\kappa - \kappa^0) + \gamma(\tau - \tau^0) da \tag{6.25}$$

where s, κ and τ are the arc length, curvature and torsion of the curve at some time t, while, s^0, κ^0 and τ^0 are the corresponding values at the rest state. The integration is taken over the whole curve Ω in the material coordinates (a). Figure (6.16) shows how these values are calculated from the described model.

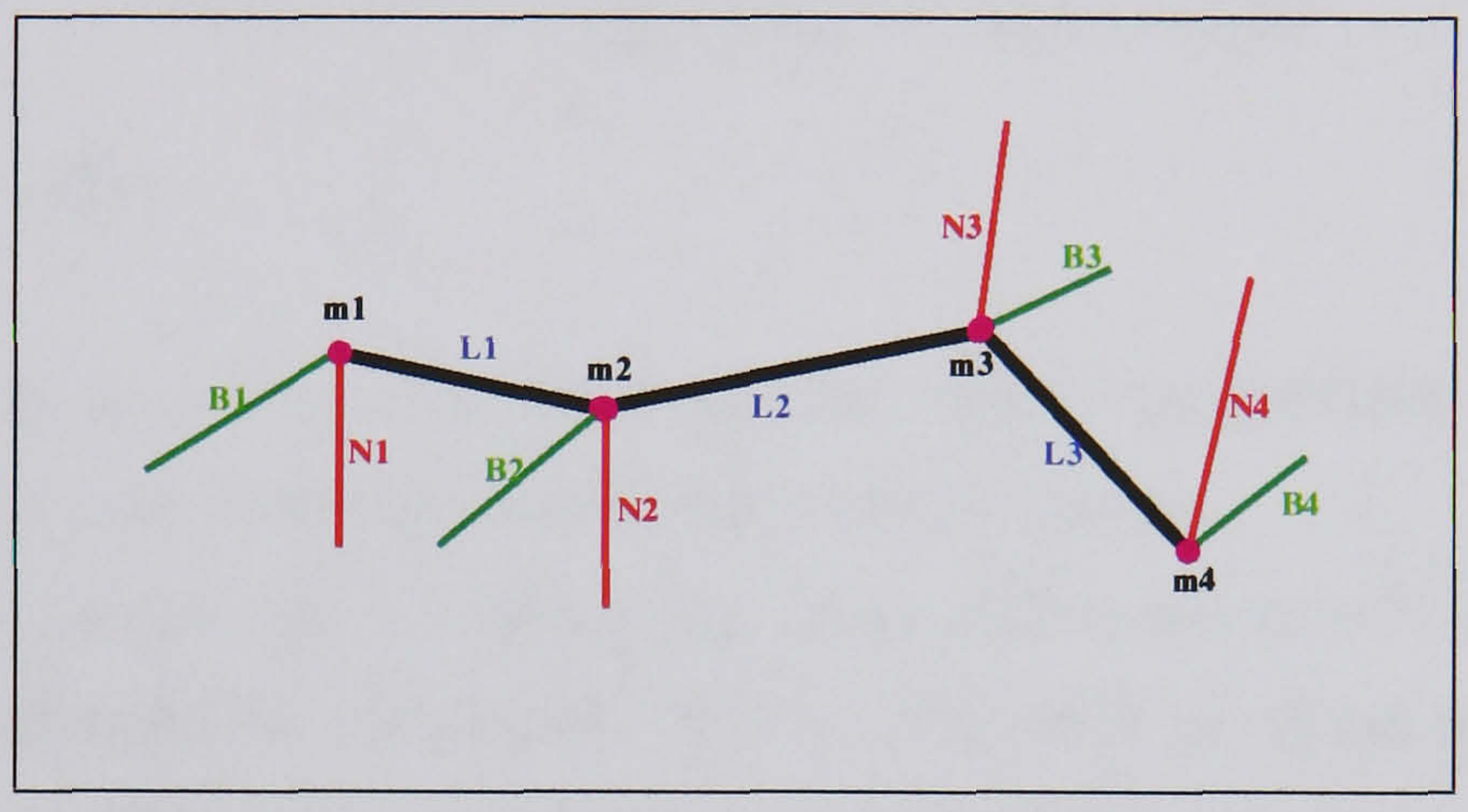


Figure 6.16: Differential geometry of the curve.

The arc length of an element is equal to the length of the vector between two mass points $\mathbf{L_i}$:

$$s_i = \mid \mathbf{L}_i \mid \tag{6.26}$$

The normal vector $\mathbf{N}$ at a mass point is calculated as :

$$\mathbf{N}_i = \frac{\mathbf{L}_{i-1} \times \mathbf{L}_i}{s_{i-1} * s_i} \tag{6.27}$$

The curvature at any point is calculated as the magnitude of the curvature of the circle passing through m_{i-1}, m_i and m_{i+1} :

$$\kappa_i = \mid \frac{2.\mathbf{N}_i}{\mid \mathbf{r}_{i+1} - \mathbf{r}_{i-1} \mid} \mid \tag{6.28}$$

where $\mathbf{r}_i$ is the vector position of point (i).

The curvature at the end points are assumed to be equal to the inner adjacent point's curvature- i.e. $\kappa(1) = \kappa(2)$. The binormal vector $\mathbf{B}$- orthogonal to both the normal and the tangent plane- is calculated as follows:

$$\mathbf{B}_i = (\mid \mathbf{r}_{i+1} - \mathbf{r}_{i-1} \mid^2 - s_{i-1}^2).(\mathbf{L}_{i-1} - s_{i-1}^2.\mathbf{L}_i) \tag{6.29}$$

Finally, the torsion is the rate of change of the binormal vector [73].

The energy is represented as a variational derivative of equation (6.25). This will yield the following expression:

$$\delta\xi(\mathbf{r}) = A.\mathbf{r} - \frac{\partial \mathbf{r}}{\partial a}(B\frac{\partial \mathbf{r}}{\partial a}) + \frac{\partial^2 \mathbf{r}}{\partial a^2}(C\frac{\partial^2 \mathbf{r}}{\partial a^2}) \tag{6.30}$$

where: $A = \alpha(s - s^0)$.
$B = \beta(\kappa - \kappa^0)$
$C = \gamma(\tau - \tau^0)$ α, β *and* γ are the material elastic properties corresponding to resistance of stretching, bending and torsion respectively.

Equation (6.30) is discretized using the finite difference method to yield a set of linked ordinary differential equations. The coefficients of these equations produce the stiffness matrix representation of this energy. This matrix for a model of four mass points is as follows:

$$\begin{bmatrix} A_1 + \frac{2B_1}{h^2} + \frac{6C_1}{h^4} & \frac{-B_1}{h^2} - \frac{4C_1}{h^4} & \frac{C_1}{h^4} & 0 \\ \frac{-B_2}{h^2} - \frac{4C_2}{h^4} & A_2 + \frac{2B_2}{h^2} + \frac{6C_2}{h^4} & \frac{-B_2}{h^2} - \frac{4C_2}{h^4} & \frac{C_2}{h^4} \\ \frac{C_3}{h^4} & \frac{-B_3}{h^2} - \frac{4C_3}{h^4} & A_3 + \frac{2B_3}{h^2} + \frac{6C_3}{h^4} & \frac{-B_3}{h^2} - \frac{4C_3}{h^4} \\ 0 & \frac{C_4}{h^4} & \frac{-B_4}{h^2} - \frac{4C_4}{h^4} & A_4 + \frac{2B_4}{h^2} + \frac{6C_4}{h^4} \end{bmatrix} \tag{6.31}$$

This matrix depends on the current state of the curve- its arc length, curvature and torsion- and is calculated every time step. The mass matrix is identical to (6.24), while damping is done on the mass point in this method, so D looks as follows:

$$\begin{bmatrix} d1 & 0 & 0 & 0 \\ 0 & d2 & 0 & 0 \\ 0 & 0 & d3 & 0 \\ 0 & 0 & 0 & d4 \end{bmatrix} \tag{6.32}$$

When we implemented this method, we had to choose the α, β and γ by trial and error. α has been given high values to simulate the resistance of the surgical tool to stretching, while β and γ were given low values to allow the model to bend and twist when it collides with the wall. However, we were not able to achieve a reasonable simulation of the surgical tool behaviour due to the high elasticity of this approach. These results are consistent with findings reported later by Terzopoulos in [175].

6.7.2 Implemented method

The method we use in our surgical training system is the hybrid approach described in section (6.5.2.4). Under this method the 1D tool is modelled as a set of mass points with two position components. The first position component is the *reference component*, which is related to the body frame. The body frame is affected by affine transformations due to the rigid motion of the body frame relative to a fixed world frame. The second component is the *elastic component*, which represents the deformation of the body from the reference component. Figure (6.17) shows the representation of the tool under this method.

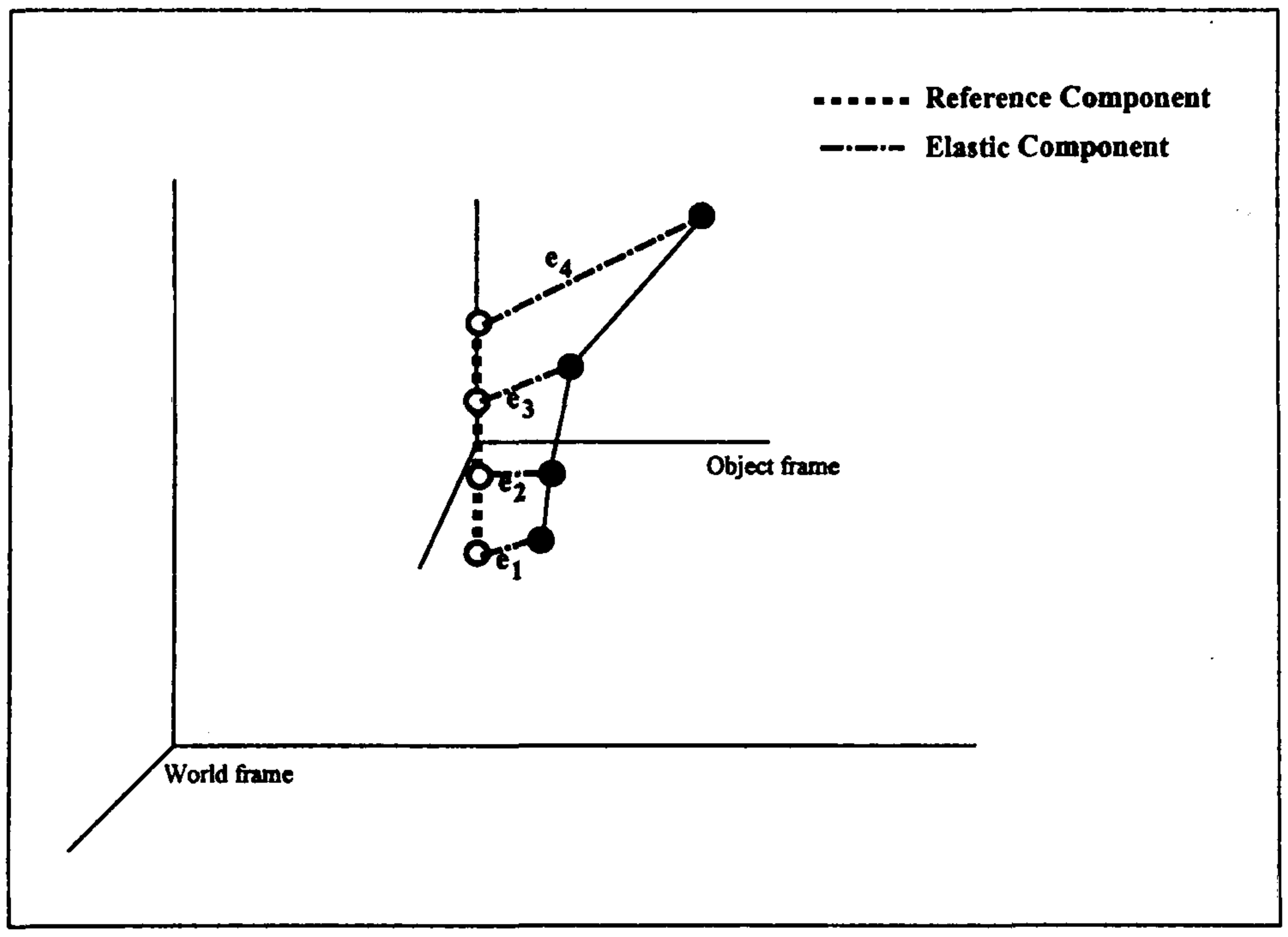


Figure 6.17: Hybrid model: tool representation

The elastic dynamics module calculates the elastic component of the model. The overall position of the body is obtained by adding the reference component subjected to the current affine transformation and the elastic component.

Internal energy is represented as a linear weighted function as mentioned earlier. The variational derivative of equation (6.17) applied to the curve case- i.e. taking p=2 and d=1- yields the following energy function:

$$\delta\xi(\mathbf{e}) = w_0.\mathbf{e} - \frac{\partial \mathbf{e}}{\partial a}(w_1\frac{\partial \mathbf{e}}{\partial a}) + \frac{\partial^2 e}{\partial a^2}(w_2\frac{\partial^2 \mathbf{e}}{\partial a^2}) \tag{6.33}$$

where *w0*, *w1* and *w2* are the weights of the function. Discretization of equation (6.33) will yield the coeficient of the stiffness matrix; for a model with N=4 this matrix is as follows:

$$\begin{bmatrix} w_{01} + \frac{2w_{11}}{h^2} + \frac{6w_{21}}{h^4} & \frac{-w_{11}}{h^2} - \frac{4w_{21}}{h^4} & \frac{w_{21}}{h^4} & 0 \\ \frac{-w_{12}}{h^2} - \frac{4w_{22}}{h^4} & w_{02} + \frac{2w_{12}}{h^2} + \frac{6w_{22}}{h^4} & \frac{-w_{12}}{h^2} - \frac{4w_{22}}{h^4} & \frac{w_{22}}{h^4} \\ \frac{w_{23}}{h^4} & \frac{-w_{13}}{h^2} - \frac{4w_{23}}{h^4} & w_{03} + \frac{2w_{13}}{h^2} + \frac{6w_{23}}{h^4} & \frac{-w_{13}}{h^2} - \frac{4w_{23}}{h^4} \\ 0 & \frac{w_{24}}{h^4} & \frac{-w_{14}}{h^2} - \frac{4w_{24}}{h^4} & w_{04} + \frac{2w_{14}}{h^2} + \frac{6w_{24}}{h^4} \end{bmatrix}$$

where w_{ij} is the ith coefficient of mass point (j). This matrix is constant and depends on the weights of the energy function and the discretisation of the model. This reduced the amount of computations needed compared to the primal method. The mass and damping matrices are similar to the ones used in the primal model.

Again, the weights of the energy function were chosen using trial and error, however we were more successful in finding values that approximated the surgical tools' behaviour. This is due to the partial rigidity of this approach, which makes controlling the stretch effect of the model when pull or push forces are applied easier.

Equations (6.18) to (6.20) are implemented in the mechanical formulation component of our model. We used the NAG C library to solve these equations in order to ensure the accuracy of the results. In particular we use the (d02pdc) ordinary differential equation solver, which solves initial value problems and uses Runge-Kutta method to give a solution for the next time step. These calculations are performed on the server using the C language, thus efficiency is provided for. Since most of our server application is implemented in Java, we used the Java Native Interface (JNI) to communicate between the simulation parts- in Java- and the numerical solver- in C. The JNI is a set of API's provided by the Java language, which can be used to access fields and methods in Java classes from applications written in C++ or C [7]. These methods are used to read the state of the object in the previous time step- i.e. $\mathbf{e}$, $\dot{\mathbf{e}}$, $\mathbf{v}$ and $\mathbf{w}$. The NAG routine is set with these initial conditions and the function is called to calculate the position of the model points in the next time step. Equation (6.20) is the function solved by this routine. The results of the NAG routine are then used to update the Java classes by calling the proper JNI functions. The rigid part of the motion, governed by equation (6.18) and equation (6.19) are solved within the Java classes- since they involve simple calculations. Appendix (B) contains more implementation details of this method.

This model is affected by two types of forces: user applied forces and collision forces. The user applied forces are received from the client side whenever they change their value- i.e. when the user interact with the virtual scope. Meanwhile, collision forces are calculated on the server side based on the last position of the surgical tool. The next section will discuss the type of collision detection and collision response techniques. Then we will give details of the methods we use in our system. Section (6.7.4) will look at the architecture of the Java classes used in our physically-based model.

6.7.3 Collision detection and response

One of the main constraints required in virtual environment, simulation and animation applications is *non-penetrating* of objects. A non-penetration constraint ensures that objects under motion do not walk through each other, as well as ensuring a realistic grabbing effect of the objects in the world. This involves solving two problems. The first is a geometrical problem, with the aim to report all contact between objects in the modelled world- known as collision detection. After that, we need to solve a dynamics problem to anticipate the changes in the state of the object as a result of the impact- called collision response. Both problems have been extensively investigated in different computer disciplines with different emphasis. Since we are concerned with virtual environment and simulation applications, we will review the requirements of this particular constraint within these applications.

In both these applications, objects undergo different types of dynamic activities due to user manipulation or internal behaviour. Thus, we are solving a dynamic version of the collision detection problem. Furthermore, the speed of collision detection is an important issue. The third concern in such applications is that the object responds realistically to this contact. This means that we are interested in knowing the contact points and their normals- i.e. collision determination is required rather than collision detection. In the next section we will review collision detection techniques suitable for virtual environments, then we look more closely at the solution we adopted. Then in the following section, we report the collision response options.

6.7.3.1 Collision detection

The aim of a collision detection algorithm is to explore the current geometry to report all contact between different objects or in some cases between the object and itself. In general, collision is either done on volumetric data- called volumetric collision detection- or on polygonal surfaces- known as surface collision detection. In the latter method, collision detection is done by investigating for edge, face and point interference. As the number of objects increases and the complexity of the objects increases, this direct approach becomes inefficient and unacceptably slow. Hence, a lot of techniques have been utilised to reduce the amount of direct face-edge checking needed. In some cases bounding regions- or a hierarchy of bounding regions- are fitted around parts of the object. A quick checking of overlapping bounding regions will eliminate unnecessary faces and edges, hence reducing the actual tests [48, 101]. Following the same philosophy, octrees have been used as well, such that faces are only tested if their parents intersect [125, 159]. Other techniques depend on tracking the distance between the objects against a minimum threshold [159]. Others utilise the inside/outside function of the implicit functions to check for penetration of convex faces [125, 65]. Finally, finding separating planes between two objects has been used to detect interference [159, 101].

Meanwhile, in volumetric collision detection a binary volume classifying the elements of the volume as either inside or outside the object is used. Volume elements- or voxels- of the different objects are investigated to check for overlap. Volumetric collision detection is faster than the surface method for complex and large objects. However, collision response is difficult since information about the normal direction to the collision surface is not available under this technique.

When dealing with dynamic objects, the above techniques can prove inefficient, since bounding box, volumetric data and octrees need to be rebuilt when the object changes shape or orientation. In some cases, temporal coherence is utilised to dynamically change the bounding hierarchy rather than rebuilding it [48, 92].

For our application, we chose a dynamic interactive collision determination technique which can handle multiple polygonal models. The technique has been implemented as a set of API's provided by the University of North Carolina, and is called V-COLLIDE. V-COLLIDE utilises a hierarchical approach to swiftly determine overlapping objects. Then an oriented bounding boxes (OBB) tree structure is used to determine the actual collision. V-COLLIDE unifies two tools to meet the requirements of a virtual environment. First, multiple-body collision detection is handled by I-COLLIDE, where bounding boxes around the different objects are used to determine overlaping objects. The bounding boxes are projected onto the x,y,z axes and a sorted list of the bounding boxes is created. The list is swept to determine pairs of overlapping objects, which is sent to the next stage. In the second stage, RAPID- a collision determination algorithm- finds polygons which collide from the two given objects subjected to affine translation and rotation. Hence, V-COLLIDE handles rigid body motion without the need to rebuild the OBB [92]. Scaling is not handled since it requires rebuilding the OBB tree. For the same reason, objects are assumed to be rigid and shape deformation is not directly handled [14]. This limitation is overcome by deleting the object model when deformation occurs and building a new one. For small models this solution is tolerable. In addition, V-COLLIDE allows the application to control the objects that are checked for collision by activating/deactivating individual object status or a pair of objects status.

During the elastic modelling of the tool, we use the 1D representation of the tool. On the contrary, during collision detection the 3D polygonal model of the tool is used. This is due to the fact that using the nodal representation for collision detection is not accurate. At any point of time, the collision detection is activated for one surgical tool at a time and the vessel. Collision detection between the surgical tools is always inactive.

6.7.3.2 Collision response

When collision occurs between two objects, the state of the objects is expected to change accordingly. Rigid objects are expected to change velocity and acceleration, while highly elastic objects exhibit shape deformation. Objects in the middle of this range may exhibit both reactions. Collision response is responsible for computing these changes and enforcing them on the object. Collision response might act directly on the object- changing the velocity and acceleration of it- or it might impose some forces on the system, that will lead to the same effect eventually. Accordingly, collision response techniques are classified into two classes: penalty methods and analytical methods.

Penalty method techniques are similar to adding spring forces to the system at the contact points. The force added is proportional to the stiffness of the spring. As the speed of collision and the weights of the colliding objects increase, the spring fails in preventing penetration. Furthermore, to correct this situation by increasing the stiffness of the spring leads to ill-conditioned numerical systems- as mentioned before. A slight variation of this method is the *repulsive field* technique. Again, forces are added to the system to prevent penetration, however these forces increase when the distance between the two objects decreases. Hence, it prevents the collision from happening in the first place [183, 28]. The advantage of the penalty method is its simplicity and applicability to wide range of object behaviour- e.g. rigid, non-rigid and articulated bodies. This method has been used to enforce non- penetration in several works- e.g. [175, 137]. Although the result of applying this method might look plausible, because it is defined in terms of position rather than dynamics, its correctness is not a proven feature. In addition, it is desirable in some applications to allow the collision to occur and then respond according to the force of collision. For instance, if we would like to simulate possible complications that might happen in an endovascular procedure, we would need to compute the collision force between the surgical tool and the vessel wall and decide according to the strength of collision and the strength of the wall if penetration would occur leading to bleeding complications. In this case, the repulsive field method will not meet the requirement.

On the contrary, analytical methods (or reaction constriants) are based on dynamic laws. They are correct but not necessarily easy to compute. Analytical methods compute impulse forces when collision occurs. Impulse forces are a special type of force which occur over an infinitely short time interval. Thus, impulses instantaneously change the velocity of the object. To calculate the impulse forces, the conservation of momentum law or non-penetration constraint can be utilised. Impulse forces are infinite forces, thus other finite forces in the system are not considered when impulse forces are applied. Baraff [27] further classifies the analytical method in the case of multiple contact points to propagational and simultaneous response. In the propagational method each collision contact is resolved independently, while a simultaneous method will calculate all the contact impulses together. Methods of calculating the impulse forces for different types of objects are presented in [101, 125].

Furthermore, implicit surfaces have been proven efficient in the case of flexible objects. Desbrun proposed using an implicit elastic surface that is detected between two colliding objects to compute the required response forces [69, 54]. Essa uses the stress- strain relationship of the implicit function- that is used to represent the object- to compute the forces needed to prevent penetration [65]. Baraff and Witkin use the same concept, however they discretise the contact surface to a mesh of triangles [30, 186].

In our training simulator, we utilise the analytical method to generate collision forces. The collision determination function will create a list of all contact points, then the propagational method is used to generate the appropriate forces for multiple collision contact. The elements of the contact list consist of a vertex number from the 1D model of the surgical tool and a surface index from the 3D triangular mesh of the vessel. The velocity of the contact point ($\dot{\mathbf{x}}$) is used to determine whether collision has happened or not, by examining the component value of this vector in the normal direction of the vessel surface. Thus, when $\mathbf{v}_n = \dot{\mathbf{x}}.\mathbf{n}$ (where $\mathbf{n}$ is the normal to the contact surface pointing outwards) is negative, the mass point is advancing in the direction of the surface and collision has occured. In this case, impulse forces are calculated using the following equations [29]:

$$\mathbf{F}_{collision} = \frac{-(1+\varepsilon).\mathbf{v}_n}{\frac{1}{m} + \mathbf{n}.(I^{-1}.(\mathbf{q}\times\mathbf{n}))\times\mathbf{q}} \tag{6.34}$$

where ε is the coefficient of restitution [0-1], which controls how elastic is the collision; $\mathbf{q}$ is the local position of the contact point relative to the material coordinates; I^{-1} is the inverse of the inertial tensor of the object.

Meanwhile the impulse torque is : $\mathbf{q} \times \mathbf{F}_{collision}$

The dynamical system is then reset and the new status of the object is calculated based on these impulse forces. The magnitude of the total collision forces of the tool is then passed to the client side to display an appropriate colour intensity for the force feedback. Once collision has been detected and calculated, the dynamical system is returned to the status before collision to resume the motion.

6.7.4 Classes and communication

In section (4.6.2), we presented a schematic diagram- Figure (4.16)- of the layers of our training system and the messages passed between these layers and the server and client sides. In this figure we included our simulation layer, however we gave no details about the content of this layer. In this section, we will examine this layer more closely.

As mentioned before, the blood vessel, the surgical tools and the user are the contents and actors of our virtual environment. These three components form the basis for the simulation layer. Thus, classes in this layer are divided into four categories: elastic object, vessel, actor and basic classes. The basic classes provide the essential utilities for the simulation layer- e.g. vector, rotation, matrix, tensor, .. etc. Actor classes are responsible for detecting user actions on the client side, send them to the server and interpret them on the server side. Meanwhile, vessel classes are responsible for setting the initial information about the vessel position and orientation on the server side. They are also responsible for handling some of the functionality of the distance and the CT slice selection tools- described in section (5.4). Finally, elastic object classes include the object representation class, hybrid energy classes, collision classes and rigid and elastic dynamics classes. These classes utilise the Java interface feature- i.e. defining abstract methods in an interface that is implemented differently by different classes- to standardise the communication between the different components of the elastic model. Thus, when we tested different PBM techniques, all we needed to provide is the new implementation for the new component without affecting the other classes.

6.8 Discussion

In our PBM we simulated the elastic behaviour of the surgical tools. We were able to provide a component-based generic solution because of the design we presented in section (6.6). Thus, we were able to experiment with several techniques to choose the most suitable for our type of behaviour. In addition, by adopting a 1D model of the modelled surgical tool we reduced the amount of computations needed as well as the network communication. Our guide wire model contains 14 nodes, while our catheter model contains 17 nodes.

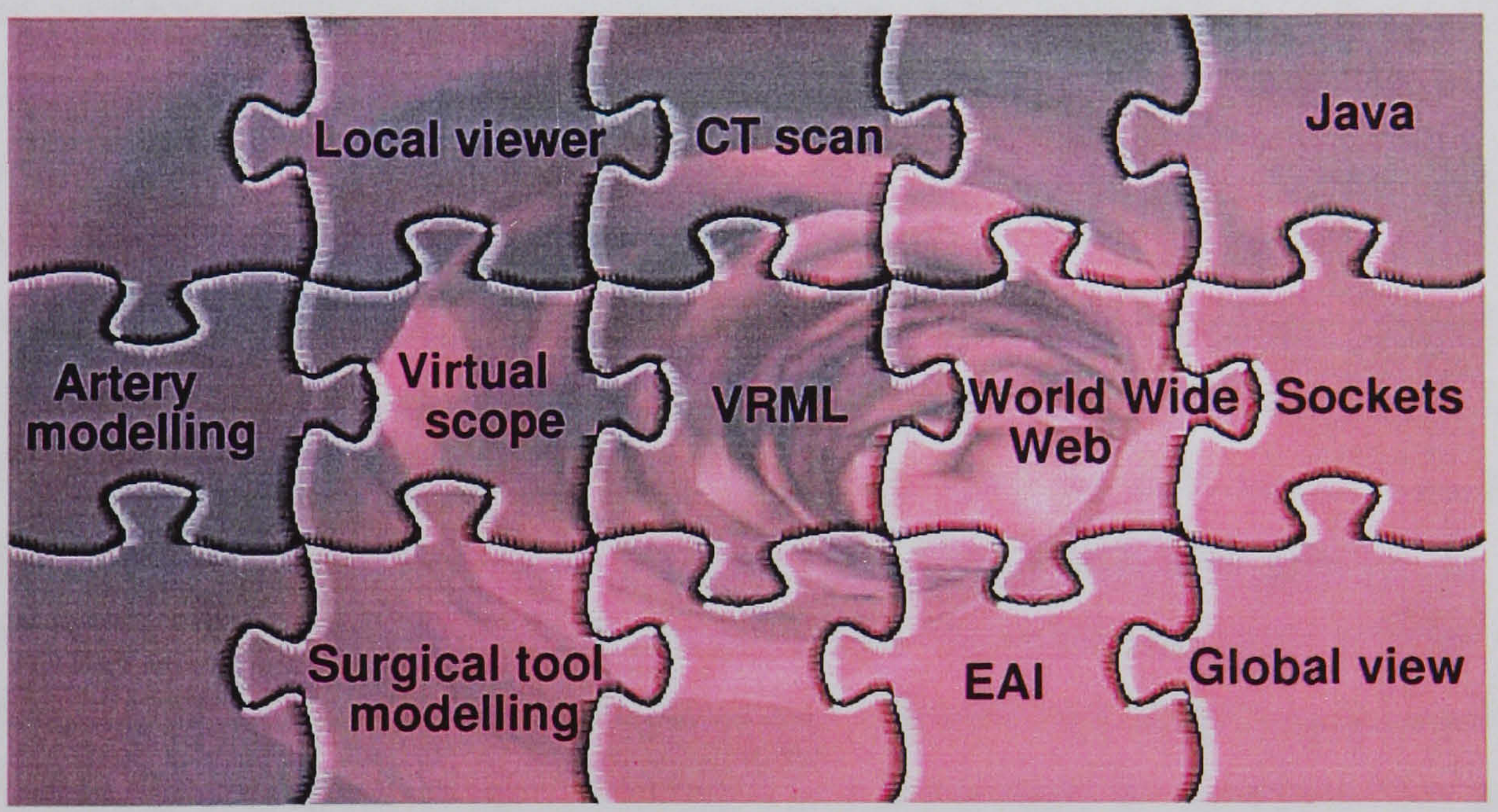
The utilisation of a well established numerical solver such as NAG library assures as far as possible the accuracy of the numerical calculations. On the other hand, the fact that it is a C based library ensures the efficient execution of the computational model, since the debate about the efficiency of Java for numerical purposes is still on going. Currently, efforts to provide Java-based numerical solutions are in their early stages - e.g. Java Ultimate Math Package [10] and Java Numerical Library [13]. However, a number of issues exist concerning the concepts, capabilities and functionality of the Java language which hinders the efficient implementation of numerical computation in Java- see [5] for a discussion of these issues and suggested solutions. The National Institute of Standards and Technology (NIST) has been experimenting with a small set of problems to maximise the performance of Java solutions taking into account the memory model adopted by Java. The result of these experiments showed that Java exhibits half the performance of the corresponding C and FORTRAN solutions.

Our simulation is able to perform the calculations for one time step in 142 msec on average with a 402 msec maximum time and 108 msec minimum time. These measurements are taken when the server is running on an SGI *Indigo*^{2} machine (200 MHZ IP22 Processor, R4400, 16 Kbytes cache and 96 Mbytes memory).

The fact that the parameters used in the physically-based model- e.g. mass, stretching and bending factors- are all arbitrary, and the simplification made in section (6.3.3), reduces the accuracy of our model. A more realistic model of the behaviour of the tool using more accurate and stable methods such as FEM is not afforable in a real time application such as ours. As we have seen from the medical applications reviewed in section (2.2), FE calculations can take up to 24 hours to reach a solution.

Chapter 7

WebSTer evaluation



7.1 Evaluation process

Now that we have described the different components of our surgical training system and discussed the effect of each component on the others, we would like to step out of the details and examine the overall picture. In this chapter we attempt to answer the second question in our hypothesis list- described in chapter (1): *how realistic, accurate and useful is the final system produced?* Hence we would like to assess our proposed solution in terms of its success in achieving the goal- i.e. training radiologists on interventional radiology procedures. In the virtual reality field, three methods have been used to appraise the technologies used in a virtual environment: physiological, performance and subjective measurement. The physiological tests measure changes in the participants' physiology- e.g. heart beat, eye scan and muscle tension- as they experience the virtual environment.

Meanwhile, the performance tests assign tasks to the participants and measure their performance in achieving this task. Performance can be measured by the time taken to complete the task, the accuracy of the achieved task or the number of trials taken to complete the task. The goal of the performance test affects the method used to conduct it. For example, when the aim is to measure the effect of a certain VR technology on the performance of the participant in the virtual environment, the test would compare the performance criteria within different settings of this environment. On the other hand in training systems, the aim is to test the transfer of skills from the virtual environment to the real world. In section (2.1.2) we saw examples of such performance tests which serve a general VR purpose. This technique has been used in [97] to assess the suitability of VR in transferring different levels of surgical skills for minimally invasive surgery. In this work basic skills such as planar motion, rotation and spatial relations have been tested with participants with different experience by measuring the time of task completion. Similarly, in the Imperial Collage of Medicine at St. Mary's the MIST VR laparoscopic training system- reported in section (2.2.6)- has been evaluated to validate the scoring system used to assess surgeons. Simple and complex tasks has been quantified and efficiency of movement, speed and errors were collected by monitoring and analysing the path taken [167]. Another method has been used by HT medical to evaluate their intravascular catheterisation system [34]. They train two groups of medical students to perform venipuncture: one with the conventional plastic arm and the other using the VR system. Each student was allowed two attempts to perform the task and was video-taped. A questionnaire has been used to measure the degree of confidence of the trainee after the training period.

Performance tests are more suitable for systems which have clear quantitative criteria. However, when assessing prototype virtual environments and when the goal of the appraisal is a subjective criteria- as in the case of measuring presence in VR systems, then subjective methods are more suitable. Subjective methods include questionnaires, verbal reports, interview and rating scales [32]. We will use this method to asses our proposed surgical training system, since this system is still in its early stages for a more formal and quantitative evaluation. We propose to use a questionnaire which includes a rating scale in order to appraise the different aspects of our solution. The goal of this assessment process is to pin point aspects that are useful in our surgical training, identify the limitations of the other aspects and gather suggestions for future developments. In order to maximise the benefit, we choose our audience such as that to cover a wide range of medical experience.

The evaluation process consisted of two sheets: instruction sheet and evaluation sheet- see appendix (D). This sheet opened by explaining the purpose of conducting this evaluation to the participant. The instruction sheet takes the participants through the features of the system and asks them to refer to the evaluation sheet at the end of each section to give a score and to add comments. This sheet has been divided into the following four sections:

- **Verbal instructions and demonstration**

  The participants are asked to go through the web pages to assess the suitability of using the web multi-media to support the early stages of training.

- **Representation skill**

  The participants are given instruction on how to use the CT selector for interactive display of CT scans as well as viewing the slices consecutively. They are asked to evaluate how useful such tools are for building the representation skill of the radiologist.

- **Measurement procedure and decision making**

  The participants are given instructions on how to use the distance measurement tool and are asked to take the necessary measurements - e.g. aneurysm neck- to decide whether this case is suitable for an endovascular procedure. They are instructed to fill in the procedure evaluation form with their measurements and decision. They are asked to evaluate the functionality of the distance tool as well as the procedure evaluation form for building the trainee's decision making and procedural skills.

- **Surgical tool simulation**

  This last section instructs the participants to use the provided surgical tools to manipulate the guide wire and catheter to the renal artery. The participant is asked to evaluate the functionality and the ease of use of the virtual scope, the collision response, the behaviour of the tools and the usefulness of the viewers provided.

Meanwhile, the evaluation sheet lists the set of requirements that we target in our system. Then a table of the solutions that we provide for each of these requirements is presented. The table includes a Grade column which is a scale from 1-3 indicating poor, satisfactory and good, as well as space for comments. At the end of the evaluation sheet, the participants are asked to reflect on the potential of using such web-based applications in the medical field. This question is included to detect other areas where such technology can be useful and to sense the reaction of the medical practitioners towards such applications. The participants are asked to name a single aspect that would greatly improve this work. The aim of this question is to detect future enhancements. Finally, the participants are asked to add any general comments which were not covered by the questionnaire.

Participants in our evaluation process are:

Dr. DK, St. James's University Hospital. Radiologist with specific experience in interventional radiology.

Dr. NP, Leeds General Infirmary University Hospital. Consultant neurosurgeon with some surgical experience in vascular surgery.

Dr. NP, Pinderfields Hospital. Surgical Senior House Officer with one year experience in vascular surgery.

Dr. AB, School of Computer Studies, University of Leeds. Computer scientist with experience in medical visualization and vision.

7.2 Results

Verbal instructions and demonstration

Participants agreed that the web pages presented with such multi-media support are useful for training purposes. It has been suggested that the amount of information presented in the web pages can be increased to cover more areas such as how to use instruments and tools in the theatre room and to give definitions for the measurements required in this procedure and examples of how to measure them.

The level of the medical contents of the web pages is thought to be too simple for experienced radiologists and too complex for public. This is due to the fact that these pages are developed to illustrate what is possible rather than an official web site about AAA.

Representation skill

The interactive display of the CT slices is thought to be useful for non-radiologist and radiology trainees, as it gives the necessary link between the 3D model and the anatomical structures in the CT slices. It has been suggested to provide an additional arrow key functionality, where the user can slide one CT slice at a time. The CT slices need to be re-rendered to show more tissue types than bone. This is due to the loss of information when the data was converted from 16 bits to 8 bits. Meanwhile, the consecutive slices display is thought to be more useful to experienced radiologists.

Measurement procedure and decision making

The distance tool is thought to be satisfactory by most participants. A suggestion was made to make the cursor smaller for better positioning of the tool and to allow the cursor to sweep on the surface of the vessel. The cursor was implemented to sweep in a 2D-plan in front of the vessel because of the curved surface of the vessel which might result in losing control over the curser when it is inside the surface. Another suggestion is to have the facility to accumulate the measurements to get the total distance of a curved path.

The procedure evaluation form is thought to be a useful technique of assessing the trainee in such an application. It was suggested that the same technique can be used to assess other decision making skills involved in the procedure such as choosing the right catheter and guide wire for each case. Furthermore, the variety of models provided is thought to be good for enriching the trainee's experience.

Surgical tool simulation

The visual feedback provided is the most disputed feature in the system. The comments collected ranged from excellent to unrealistic. It has been suggested to use visual motion resistance as an alternative- where collision is represented as a slow down or complete stopping of the tool's motion.

The behaviour of the surgical tool is thought to be unrealistic and some times slow. It has been noticed that the rendering was lagging the user interaction and that caused confusion when using the tools. Also the fact that we only model the tip and forces are applied to the end of this tip made the surgical tool moving in a longitudinal axis all the time which was limiting the task achievement. In addition, some of the catheter functionality was not medically correct- e.g. catheters are not advanced once they are passed over the guide wire and the tip of the catheter shape would change during this process. In general, the difficulty of the tackled problem was appreciated by the participants and the general impression was to attempt a simpler modelling technique even if it is not very accurate.

Both the global and local views are thought to be useful during the manipulation of the surgical tools. However, one participant expressed caution about having the trainees depend on the local view when it does not correspond to the real life situation. The rendering lag is again noted to be confusing. Furthermore, one participant expressed his inability to recognise how far the tool was from the walls of the vessel from the local viewer. Thus, we searched the VR literature for explanation to this note and we found that some experiments were conducted in [84] to investigate the effect of the geometric field of view on the feeling of presence in VR systems. In this work, it has been reported that a field of view of 90° and 50° can provide a better sense of presence since it gives more information about the environment. Our initial field of view was 40° and when increased to 90° it gave a better view of the lumen.

General questions

There was an unanimous acceptance of the idea of using the web as a training environment. The participants seem to be aware of the advantages of the web in providing distance learning, training scenarios, training assessment and even procedure rehearsal of patient specific data. One of the participants suggested to apply such technology to more popular procedures such as laparoscopy procedures.

Suggestions for future enhancement of the system seem to focus on providing specialised input device instead of the mouse and haptic force feedback. In addition, one participant stressed on the importance of interactive realistic behaviour of the surgical tools.

In general, it is thought that we have attempted to tackle a reasonably complex problem and provided a good starting point to solve it. Furthermore, the system is thought to be useful in identifying difficulties involved in such application.

7.3 System testing

When we proposed our solution in section (3.4), we claimed that using the web as an environment for training will provide us with an accessible affordable solution. To verify the accessibility of the our system, we have tested it on different platforms with different capabilities. We found that the client side was completely portable across these platforms. Table (7.1) shows the platforms we tested and their configuration.

O^2 R10000	195 MHz IP32 Processor, 1024 Mbytes memory, Ethernet ec0, CRM graphics
O^2 R5000	180MHz IP32 Processor, 96 Mbytes memory, Ethernet ec0, CRM graphics
PC windows NT	Pentium II 333 MHz, 128 MBytes RAM, Fire GL 1000 Pro graphics card
Notebook	133MHz, 16 MByte RAM, PCMCIA 33.6 voice modem

Table 7.1: Platforms used to test our training system.

We have also tested the training application on the Internet with the cooperation of the Manchester Visualization Centre, University of Manchester. We ran the client side remotely on an SGI R4400 machine (200 MHZ IP22 Processor, 64 Mbytes memory, Indy 24-bit graphics card). This test demonstrated the portability and usability of our web-based training system.

Chapter 8

Conclusion and Future work

8.1 Summary

There is a need for systematic efficient methods for training medical staff. The current surgical training methods do not provide enough hands-on practice or systematic evaluation of the trainee performance. In addition they are geographically localised and expensive. Virtual reality (VR) technology has been used to provide solutions to this problem. These solutions might cure the first two limitations- i.e. practise and evaluation- but they are still localised solutions that cost even more than the traditional methods. In this thesis we have investigated a different solution which targets these short comings.

At the beginning of the thesis, we hypothesised that the combination of VR and the web can provide for the needs of a training system. Such a hypothesis is built on the fact that VR provides realistic training environment, while the web provides accessibility. We also claimed that the web can support all stages of training due to the multi-media supported. We were then able to provide a training solution by utilising a combination of VR and web technologies. The combination of these technologies was able to support some of the training system requirements. In fact we were able to deliver an example of how the web can be used to foster a complete surgical training system. We have presented web pages with multi-media tools- text, images and movies- to introduce the trainee to the procedure of interest and to provide different sources of infoŕmations. Then we provided a virtual system that the trainee can practise the procedure and gain the different skills involved. Finally, we used a combination of the web pages and tools in the virtual environment to evaluate the performance of the trainee.

The virtual simulator is able to support a number of skills needed in an interventional procedure: representational, procedural and decision making skills are well presented in this environment. Meanwhile, eye/hand coordination and psychomotor skills are only partially supported due to the lack of the appropriate I/O devices.

We have found that the web is able to support computational demanding applications via its server-based architecture. However, currently the web only supports conventional I/O devices, which might reduce its utilisation in more realistic simulators that require special peripherals. In its current state, the web cannot efficiently support real time applications especially when network communications is vast.

Despite all these limitations, we have presented an interactive simulation environment which models both the geometry and the elastic behaviour of the surgical tools. We were able to provide such a system by adopting a server-based architecture, where dynamic calculations are done on the server and a virtual interface is presented on the client side. We have reduced the computational cost of the modelling component and the network communication by simplifying the model of the surgical tool. We were able to achieve accurate and efficient modelling by utilising a well established numerical solver. In addition, we provided multiple view windows on the virtual interface that has been achieved by utilising a combination of VRML, Java and the EAI technologies. We substitute for the lack of special I/O devices by providing virtual tools and visual and audio stimuli for force feedback. Finally and most importantly, we provided a platform independent, scalable and accessible training system by choosing the web as the environment for our system.

8.2 Future work

From the appraisal process that we presented in chapter (7), we collected suggestions for future development and enhancement of the this work. Firstly, the web pages developed to support the initial stages of training can be further developed to provide a variety of levels suitable for public, trainee radiologists and experienced radiologists. The content of the pages would reflect the level of experience of the audience.

Secondly, the training simulator can be simplified to gain more interactivity rather than accuracy. It is felt that the accurate simulation of reality is not essential for training purposes as much as conveying the training messages. For example, it might be more useful for a trainee to receive an immediate error messages or warning bleep when applying excessively big forces than to get the same information delayed after going through the mechanical calculations. The appraisal process gave the message that "realism is more important than accuracy". Two approaches have been suggested to achieve this target. One is to pre-calculate the possible scenarios of a case, then use some sort of rule-based system to either select the next action based on the precalculated cases, or interpolate it based on them. The second suggestion is to simplify the surgical tool model into segments of rigid objects and attempt a more client-based approach.

Furthermore, user applied forces, collision strength, number of trials attempted to achieve a task and errors performed during the training can be detected and stored in a trainee profile which is used later for more systematic evaluation of the trainee. In addition, the force feedback and specialised input devices which mimics the surgical tools are another area which would improve the clinical usage of this system.

Finally, a more formal evaluation- perhaps using a performance method- for the skills that has been well presented in the system is needed to measure the transfer of these skills to real life. For example, using the interactive CT display with the 3D geometry was unanimously thought to be a useful tool by our evaluators. In order to answer the question "Does this tool develop the representation skill of a new radiologist?", two groups of radiology trainees would be involved in a formal evaluation process, where by one group is trained using the conventional methods and the other using this tool.

8.3 Conclusion

Providing surgical training systems on the web possesses a number of advantages to the training process of the medical staff. Technology now exists to support a prototype training application. The exponential rate of technology development and improvement in this field indicates that this support will improve in the near future. This work illustrated the utilisation of a number of current technologies to develop a surgical training which covers the various stages of the training process. Although we used surgery as an application, it shares some general training concepts that can be used to develop other training applications.

Appendix A

VRML-Java communication: Implementation details

As mentioned earlier, the communication applet is responsible for receiving commands from the server and updating the the VRML worlds of the user interface- the global and local viewer. The "simulation_client" class is the class responsible for detecting changes in the VRML worlds and updating the nodes comprising these worlds. This class utilises the External Authoring Interface (EAI) to communicate with VRML. Below is a skeleton of this class which shows its main functionality.

```
import vrml.external.field.*;
import vrml.external.Node;
import vrml.external.Browser;
import vrml.external.exception.*;
class Simulation_client extends Applet implements EventOutObserver {
public Simulation_client(DataInputStream ,PrintStream ,Applet , ...) {
   GetBrowsers();
   GetNodes();
...
}
public void interpretInput (String ) {
...
}
public void callback(EventOut ,double ,Object ) {
...
}
}
```

The first thing to note in this skeleton is that the class imports the Java packages which include the External Authoring Interface. Then the class implements the "EventOutObserver" interface which will allow the class to detect changes in

VRML EventOut fields. This interface will require the implementation of the "call-back" method which receives events when one of the registered fields change. This method should be implemented to take the appropriate action when certain events are detected. For example, when the force field in the virtual scope node in the control panel VRML world changes- indicating that the user is applying force to the surgical tool; this method will send the appropriate message together with the force value to the server side.

The constructor of this class is responsible for grabbing a pointer to the VRML worlds embedded in the same HTML page. This is done by calling the following EAI method:

browser = (Browser) vrml.external.Browser.getBrowser(Applet,"",i)

This method provides the following two parameters: the communication applet object and the slot number of the VRML world desired- such that the first VRML world embedded in the HTML page has the slot number zero.

The second task of the constructor is to grab pointers of the nodes and fields of interest in the VRML worlds. In order to get a node from a VRML world- pointed to by "browser"- the following EAI method is called:

Node node=browser.getNode(Nodename)

The "Nodename" is the name given to the node using the DEF feature of the VRML file format. EventIn's and EventOut's of this node can be accessed by calling one of the following methods:

Button1=node.getEventOut(Fieldname)

Button2=node.getEventIn(Fieldname)

EventOut's are registered in order to detect changes to their values. This is accomplished by calling the "advise" method of this event and assigning a number (n) to the event; this number is used in the callback method to identify the coming event.

Button1.advise(this, n)

The DataInputStream and PrintStream parameters of the class constructor are the input and output streams created by the client when a socket connection was opened with the server. Meanwhile, the Applet parameter is the communication applet object which is needed to get the browser pointers.

The second important method in this class is the interpretInput method which is called by the "VC_client" class when server commands are received and are not interpreted by the "VC_client" layer. This method implements the appropriate action to be taken when data is sent from the simulation engine on the server. For example, when this method receives an "UpdatePos" message, it will read the new position vector of the N control points comprising the surgical tool model. These vectors are used to update the position of the spine nodes of the Extrusion models of the surgical tool which are included in the global and local viewer VRML worlds. The position of the lowest node in the model is used to update the viewpoint position of the local viewer VRML world. In order to update the fields of any node, the appropriate EAI method is called depending on the type of the field- refer to [3] for a list of these methods. For example, the view point position is of type SFVec3f- i.e. one 3D vector- while the spine field of the Extrusion node is of type MFVec3f- i.e. multiple 3D vectors. Hence, in order to update the viewpoint value, the following method is called:

VPtrans.setValue(NewPos)

Meanwhile, in order to update the i^{th} element of the spine field, the following method is called:

GWPos.set1Value(i,NewPos)

Appendix B

Hybrid method: Implementation details

The hybrid method that we implemented in our server-based simulation engine is used to model the behaviour of the surgical tool. At each time step and upon receiving the time command from the client, the simulation engine will perform the following algorithm:

Check collision
if (ColForce > 0)
  Send ColForce to client side
Compute rigid dynamics
Compute elastic dynamics
Update object state
if (ColForce > 0)
  Reset dynamics

B.1 Check collision

This function will call the V-COLLIDE "vcCollide" method, which returns a report of the number of collision pairs which occurred between the blood vessel and the currently active surgical tool. Each pair in the list represents a triangle number in the vessel "IndexFaceSet" and a control point number in the 1D model of the tool. Before collision response is calculated, the current dynamics of the body is reset to eliminate finite forces. Each pair is passed to a collision response function where collision force is detected and set to the appropriate control point- as mentioned in page (124). The average magnitude of the collision forces is calculated and passed to the client side with the appropriate message.

B.2 Rigid dynamics

The calculation of the rigid part of the tool motion is the responsibility of the "RigidDynamics" layer (class). This class implements the following algorithm, in order to compute the velocity/acceleration and angular velocity/acceleration of the centre of mass of the object.

Compute Fv and Fw.
Compute $\dot{\mathbf{v}}$ and v.
Compute Distance.
If Distance != 0
  Send Distance to client side.
Compute $\dot{\mathbf{w}}$ and w.
Compute Angle and Axis.
If Angle > 1°
  Send Angle and Axis to client side.

Where :

- $\mathbf{Fv} = \sum \mathbf{f_i} + LE$ *where* $\mathbf{f_i}$ is the force acting on element i and LE is the linear elastic force contributed from the elastic layer- see section (B.4).
- $\mathbf{Fw} = \sum \mathbf{T_i} + AE$ *where* $\mathbf{T_i}$ is the torque acting on element i and AE is the angular elastic force contributed from the elastic layer- see section (B.4).
- $\dot{\mathbf{v}}$ is the acceleration of the centre of mass and is equal to $\dot{\mathbf{v}} = \mathbf{Fv}/m$ where m is the total mass of the surgical tool as if it is concentrated in one point.
- $\mathbf{v}$ is the velocity of the centre of mass, which is equal to $\mathbf{v}_{t+\Delta t} = \mathbf{v}_t + \frac{\Delta t}{M}.\mathbf{Fv}$
- Distance is equal to $d = \Delta t.\mathbf{v}$
- $\mathbf{w}$ is the angular velocity of the centre of mass and is equal to $\mathbf{w}_{t+\Delta t} = I_{t+\Delta t}^{-1}(I_t.\mathbf{w}_t + \Delta t\mathbf{Fw})$, where I is the inertia tensor of the object.
- $\dot{\mathbf{w}}$ is the angular acceleration of the centre of mass and is equal to $\dot{\mathbf{w}} = \frac{\mathbf{w}_{t+\Delta t} - \mathbf{w}_t}{\Delta t}$
- Angle is the new rotation angle of the centre of mass and is equal to $\Delta t.|\mathbf{w}|$
- Axis is the axis of rotation of the tool and is equal to $\frac{\mathbf{w}}{|\mathbf{w}|}$

B.3 Elastic dynamics

Meanwhile, the elastic component of the surgical tool motion is calculated in the "ElasticDynamics" class. This class calls a C function which implements equation (6.20) and uses NAG functions to solve it numerically. Calling this function from the Java file is done using the Java Native Interface (JNI) methods. First, the Java class will include the following declaration of the function in its methods list:

```
public native void calculateElastic(double time, ...)
static {
System.loadLibrary("Endoscopy");
}
```

This declaration will give the name of the library which contains the implementation of this method. On the other hand, the declaration of this function in the C file will look as follows- refer to [7] for details of how to use JNI.

```
JNIEXPORT void JNICALL Java_ElasticDynamics_calculateElastic (JNIEnv ,
jobject , jdouble time, ..)
```

The first two parameters of this function will provide a pointer to the JNI interface and the object owning this method. These two parameters are used to access methods and fields in the Java classes involved in this calculation. This method implements the following algorithm:

Set numerical parameters of the NAG function.
Set the initial values of the unknown parameters (i.e. $\mathbf{e}, \dot{\mathbf{e}}, \ddot{\mathbf{e}}$).
Call the initialisation function (d02pvc).
Call the initial value ODE solver function (d02pdc).
Set the results to the Java parameters ($\mathbf{e}, \dot{\mathbf{e}}, \ddot{\mathbf{e}}$).

The function solved by the numerical solver is :

$$\frac{d}{dt}\mathbf{e} = \dot{\mathbf{e}} \tag{B.1}$$

$$\frac{d}{dt}\dot{\mathbf{e}} = \mathbf{G_e} - \gamma\dot{\mathbf{e}} + K\mathbf{e} \tag{B.2}$$

where K is the stiffness matrix and $\mathbf{G_e}$ is the force acting on the control point- see section (B.4).

B.4 Update Object State

In this part of the algorithm, calculations are done to update the dynamical status of the surgical tool for the next time step. These calculations include computing the velocity of each control point using the following equation:

$$\dot{\mathbf{x}} = \mathbf{v} + \mathbf{w} \times \mathbf{q} + \dot{\mathbf{e}} \tag{B.3}$$

Next, calculate the linear and angular contribution force of the elastic layer in the rigid dynamics layer, where linear elastic force (**LE**) is equal to:

$$\mathbf{LE} = -\sum \mu.\dot{\mathbf{e}} - \sum \gamma \dot{\mathbf{x}} \tag{B.4}$$

Meanwhile, the angular elastic force (**AE**) is equal to:

$$\mathbf{AE} = -\sum \mu.\mathbf{q} \times \dot{\mathbf{e}} - \sum \gamma \mathbf{q} \times \dot{\mathbf{x}} \tag{B.5}$$

Then, $\mathbf{G_e}$ is calculated, where $\mathbf{G_e}$ is the force acting on an individual control point- note this equation is a simplified version of Terzopoulos's equation (6.20).

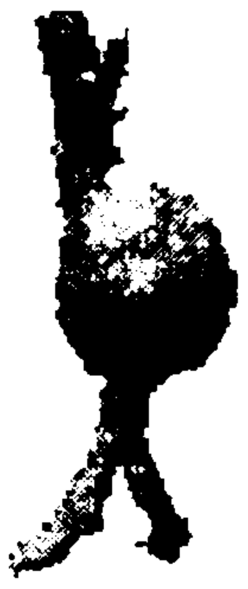
$$\mathbf{G_e} = \mathbf{f_i} - \mu \dot{\mathbf{v}} - \mu \dot{\mathbf{w}} \times \mathbf{q} \tag{B.6}$$

Finally, this function will check if the surgical tool has encountered a change in the elastic component- i.e. local deformation. If such deformation has occurred, it will send the new positions of the control points to the client side and delete and reconstruct the collision model of the tool- since our collision detection algorithm cannot handle deformation automatically- as mentioned in section (6.7.3.1). If collision has occurred at this time step, the previous state of the object before collision is returned to resume the dynamics.

Appendix C

WebSTer web pages

- Page (146) : Introduction page to the AAA pages.
- Pages (147- 150) : Verbal instructions and endovascular procedure description.
- Pages (151- 152) : Demonstration movies for the surgical tool manipulation.
- Pages (153) : Other sources of information about AAA.
- Pages (154- 156) : Description of how to use our surgical training system.
- Pages (157) : Consecutive display of CT slices to build representational skills.
- Pages (158- 159) : AAA models avialable for training.
- Pages (160) : Endovascular procedure evaluation form to build decision making skills.



Web-based Surgical Trainer

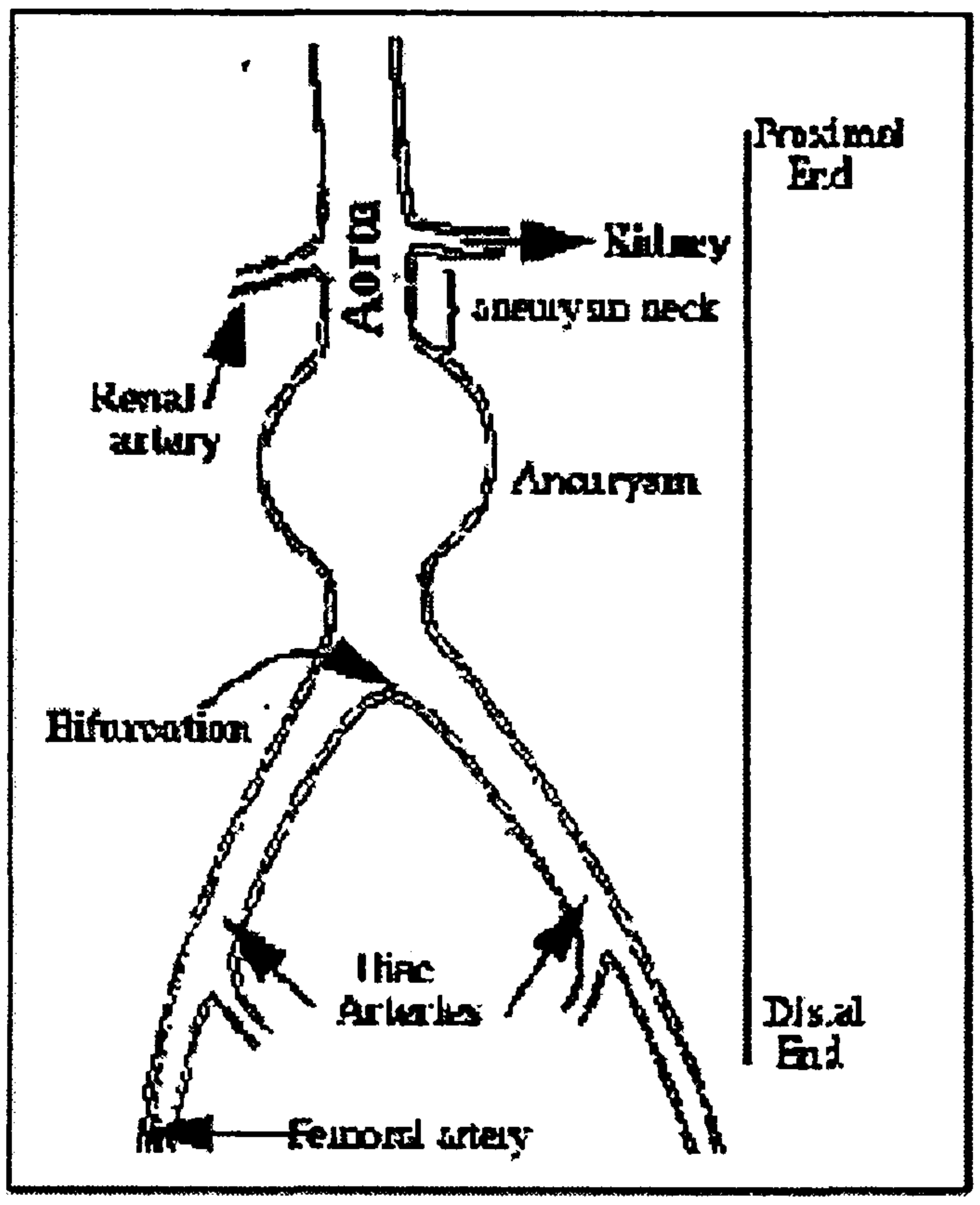
This page is part of a Phd research which investigates the suitability of using the World Wide Web environment to provide surgical training systems. The argument that we put forward is that currently the web provides multimedia technologies which can support all phases of training: verbal instructions, demonstration, practice and evaluation. Text, movies and audio streams can be used to satisfy the first two phases. Technologies such as Java, VRML, Java3D can be used to provide a simulation environment that is used to practise the procedures and evaluate the trainee. We investigate these hypotheses in the interventional radiology area and in particular in procedures that are used to treat Abdominal Aortic Aneurysm (AAA). This project is being done at the School of Computer Studies, University of Leeds, in co-operation with St. James's University Hospital.

These pages present textual information about the steps of the procedure and relevant information on the Web. Movies of surgical tool manipulation done on a plastic model of the aorta are included to demonstrate the procedure. Then we provide a training system to practise the procedure. The training system is a simplification of the actual procedure that will allow the trainee to navigate in the aorta with flexible surgical tools that mimic the real life tools- a guide wire and a catheter are available. All you need to run this system is a Java enabled browser, VRML plugin and EAI classes. The training system is based on a client-server architecture, and communication with the server is needed during the training session. This might cause the system to be slow when the network is congested. Currently, we are not able to provide a dedicated server for this application, however, a server is sometimes operated on a trial basis. We hope that in the future we will be able to provide a full time service. Please note that this is still ongoing work, and your feedback is important to us (email: nuha@scs.leeds.ac.uk , for comments).

- AAA Endovascular Procedure.
- AAA Movies.
- AAA Links.
- WebSTer: AAA Training Simulator.

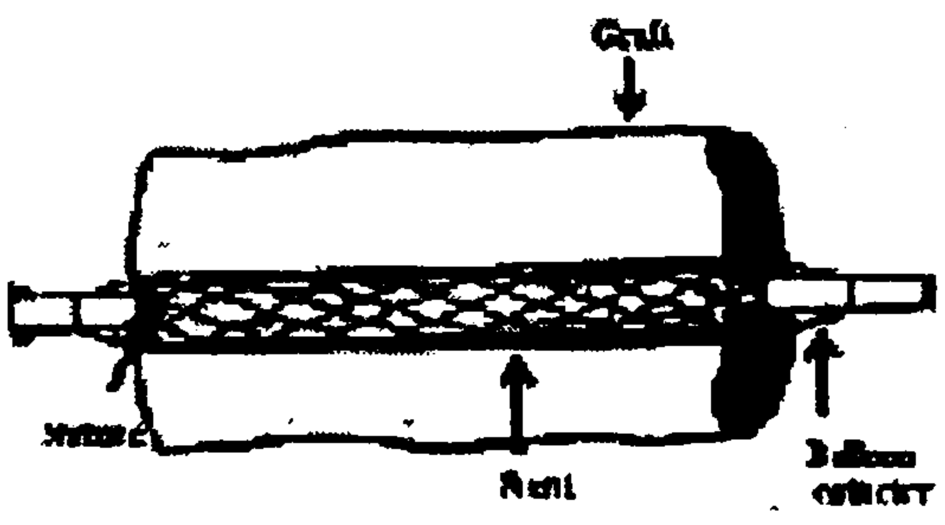
AAA Endovascular Procedure

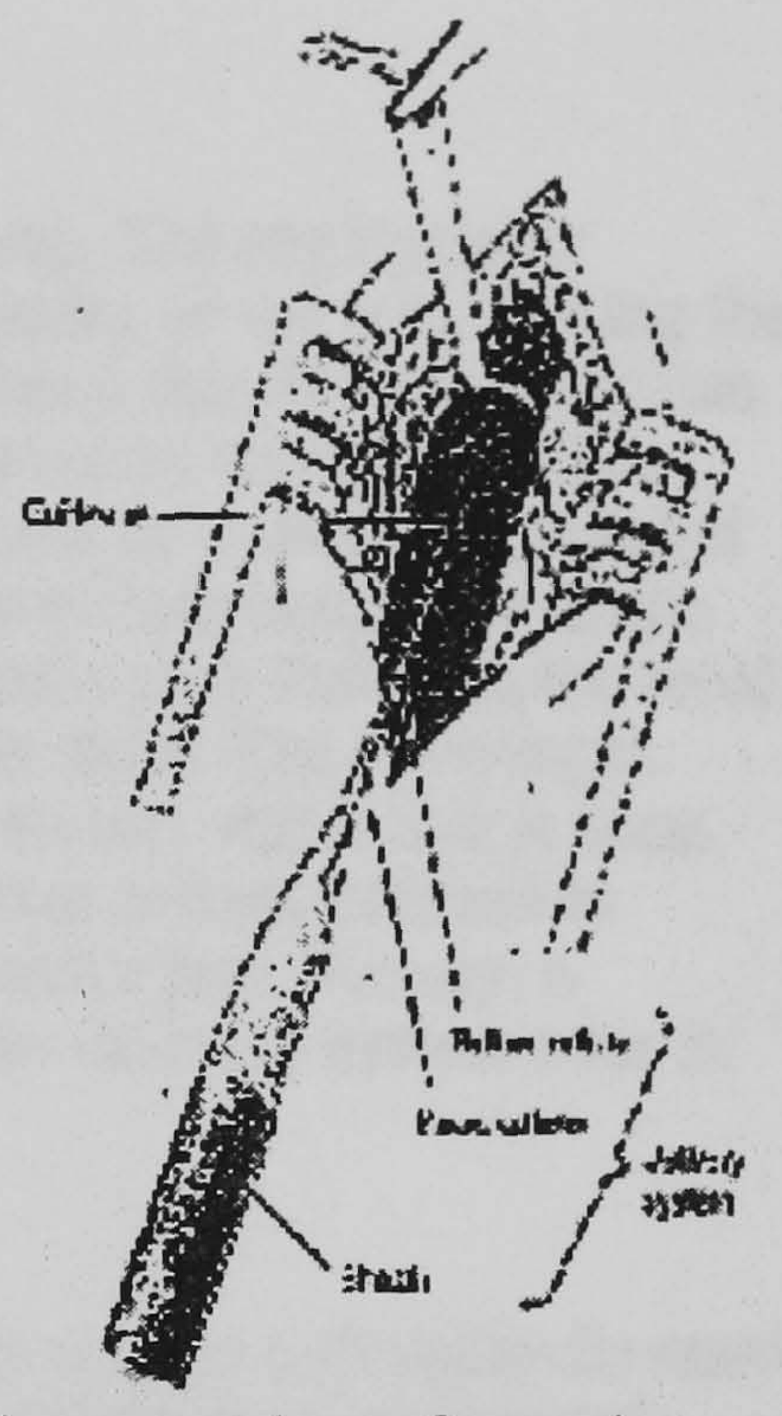
Anatomy



Equipment

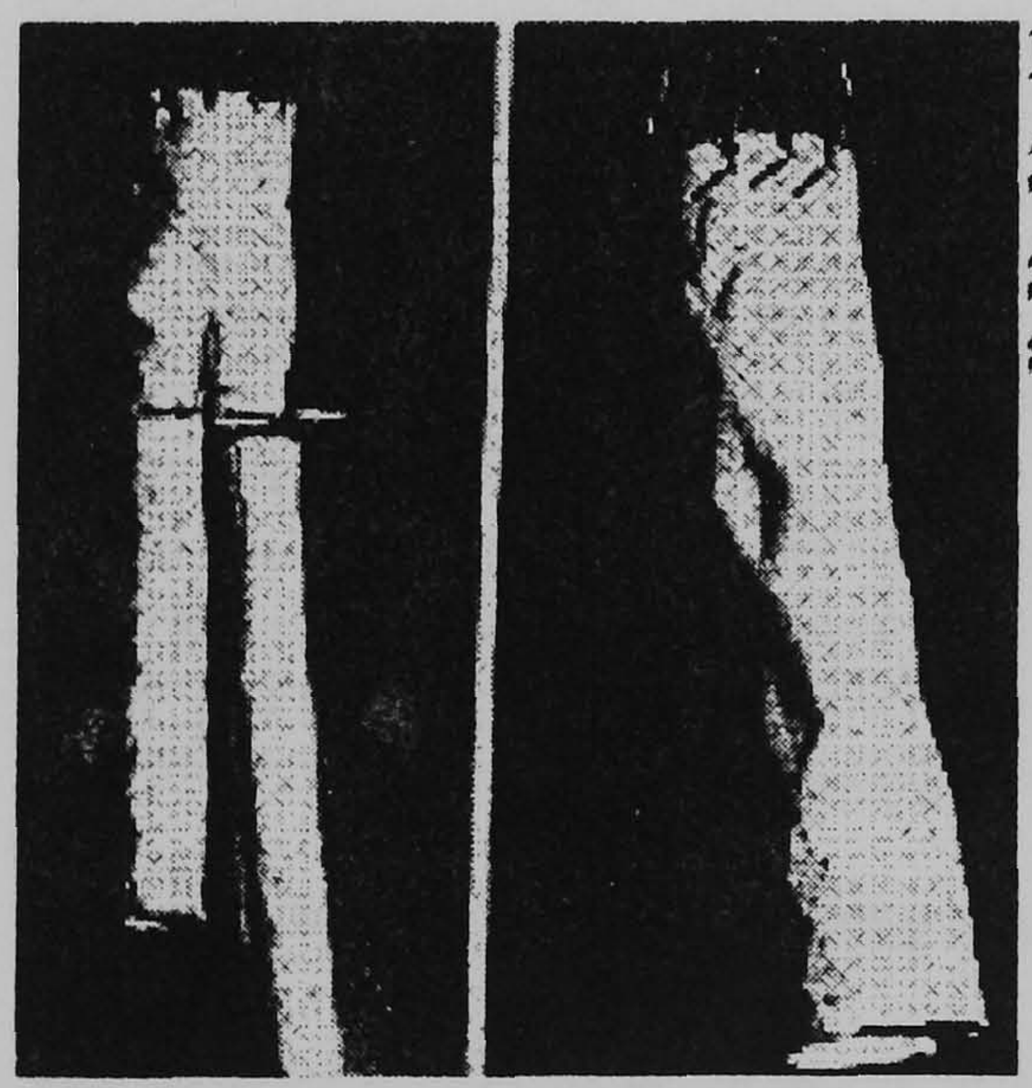
The goal of the procedure is to place a stent-graft across the aneurysm to exclude it from the blood circulation. The stent is a metal mesh (8–14 cm long and 18–25 mm in diameter) covered with a polyester fabric, which is the graft. The graft is fixed to the stent with sutures in the proximal and distal ends.





The stent-graft is part of a delivery system that consists of a pusher catheter, balloon catheter and sheath. The sheath is used to deploy the stent.

This procedure is characterised by one of two modes, which depends on the position and extent of the aneurysm. A straight stent-graft placement is done when the aneurysm is proximal to the bifurcation- which form 6% of patients. On the other hand, a bifurcated stent-graft placement is performed when the aneurysm extends to the iliac arteries.



Picture extracted from [U. Blum, et.al. Abdominal Aortic Aneurysms: Preliminary Technical and Clinical Results with Transfemoral Placement of Endovascular Self-expanding Stent-Grafts. Radiology:(198)(1):25-31,1996]; left: bifurcated stent-graft, right: straight stent-graft.

Endovascular Procedure

Step 1: Arterial access and imaging.

The surgeon exposes the right femoral artery with an incision 5-7 cm long. The angiography catheter is inserted from the same incision, the left side of the femoral artery or the arm. During the insertion of the angiography catheter some difficulties might be experienced due to a tortuous iliac artery. In this case, it is recommended to use a combination of shaped catheters and hydrophilic guide wires to facilitate access. An angiography image is taken, then a contrast material is injected and the imaging is repeated again. A digital subtracted image of both these images appears on the monitor, showing the vascular structure. The image intensifier is positioned such that both the renal and bifurcation appear on the monitor- a field of 22cm is needed to cover both. The radiologist marks the position of the renal artery and bifurcation with a tape on the screen. After this is done, the image intensifier and the patient must not be moved. A marked catheter is then inserted to measure the aneurysm neck and the distance from the bifurcation to the aneurysm. Finally, a J-shaped stiff guide wire is inserted in the right femoral artery to pass the delivery system over it.

Step 2: Insert the delivery system.

Secondly, pass the delivery system over the guide wire. Such a task is not always a straight-forward one. The surgeon incise the femoral artery to insert the delivery system. When the iliac artery is tortuous, the insertion of the delivery system is difficult: in this case the arteries can be straightened by palpating the iliac artery or applying pressure on the abdomen. If it is impossible to avoid the tortuous structures, the left side arteries may be used alternatively. Advance the delivery system until it reaches the renal arteries. Angiography images are repeated here to make sure of the position of the tool and to check that the distal suture is above the bifurcation- i.e. the length of the stent is correct. The stent is usually positioned 1-2 mm across the renal artery, that is because the end position of the stent is usually 3-4 mm lower than the start of the deployment. Then the sheath is unlocked and withdrawn to deploy the upper stent. Caution should be taken during this task not to move the stent while pulling the sheath. When the aneurysm neck is not straight, it is difficult to judge the correct position of the upper stent. In this case, angiography imaging should be taken from a perpendicular position if possible. Finally, the radiologist checks the position of the down stream stent and withdraws the rest of the sheath to deploy it. If the downstream stent needs shortening, the upstream stent is released using the pusher and the balloon catheter and the stent length is adjusted.

Step 3: Withdraw the delivery system.

The delivery system is removed by retreating the balloon catheter first. Then the delivery system is withdrawn. Caution should be taken during the withdrawal operation, so that the delivery system does not catch on the upper or down sutures, which cause the stent to migrate. If difficulty is encountered when withdrawing the tool, angiography imaging should be performed and the sutures are exposed to release the delivery system.

Step 4: Post angiography.

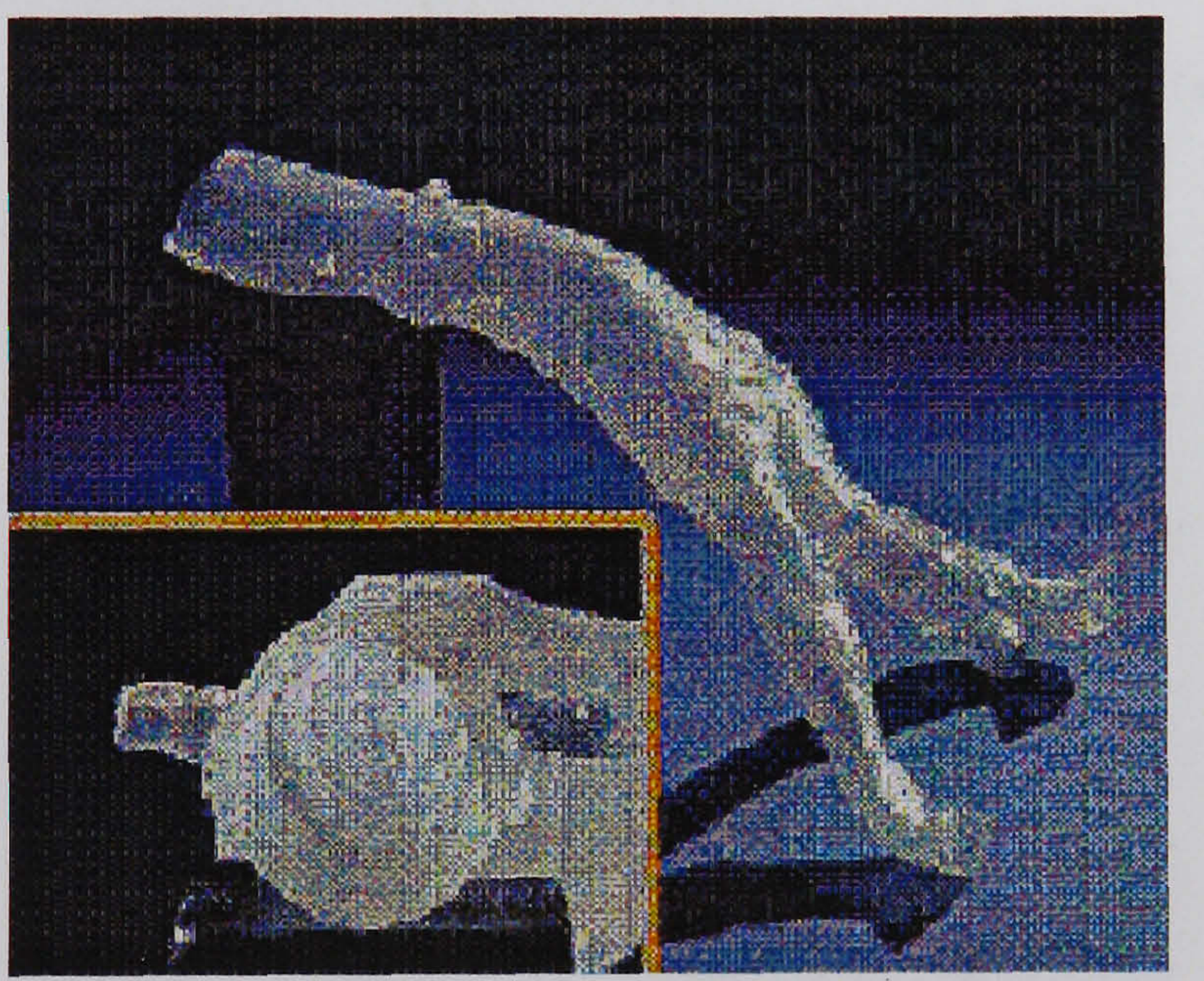
Finally, angiography images are taken again to check the position of the stent. The angiogram will show if blood leaks from the distal or proximal end as well as the reason for the leak. The table below shows the possible complications and the required remedy.

Proximal stent is **high**, covers millimeters of the renal artery	use a balloon to pull the stent down
Proximal stent is **high**, blocks the renal artery	perform open surgery
Proximal stent is **low**, blood is leaking into aneurysm	place another stent on top of the first to cover the leak
Proximal stent is **angulated**, blood is leaking	use a balloon to correct the orientation.
Distal stent is **high**, aneurysm is not covered	place another stent to cover the rest of the aneurysm
Distal stent is **low**, covers iliac artery	perform open surgery
Distal stent is **angulated**, causing retrograde leak	use a balloon to seal it.

Back to main page

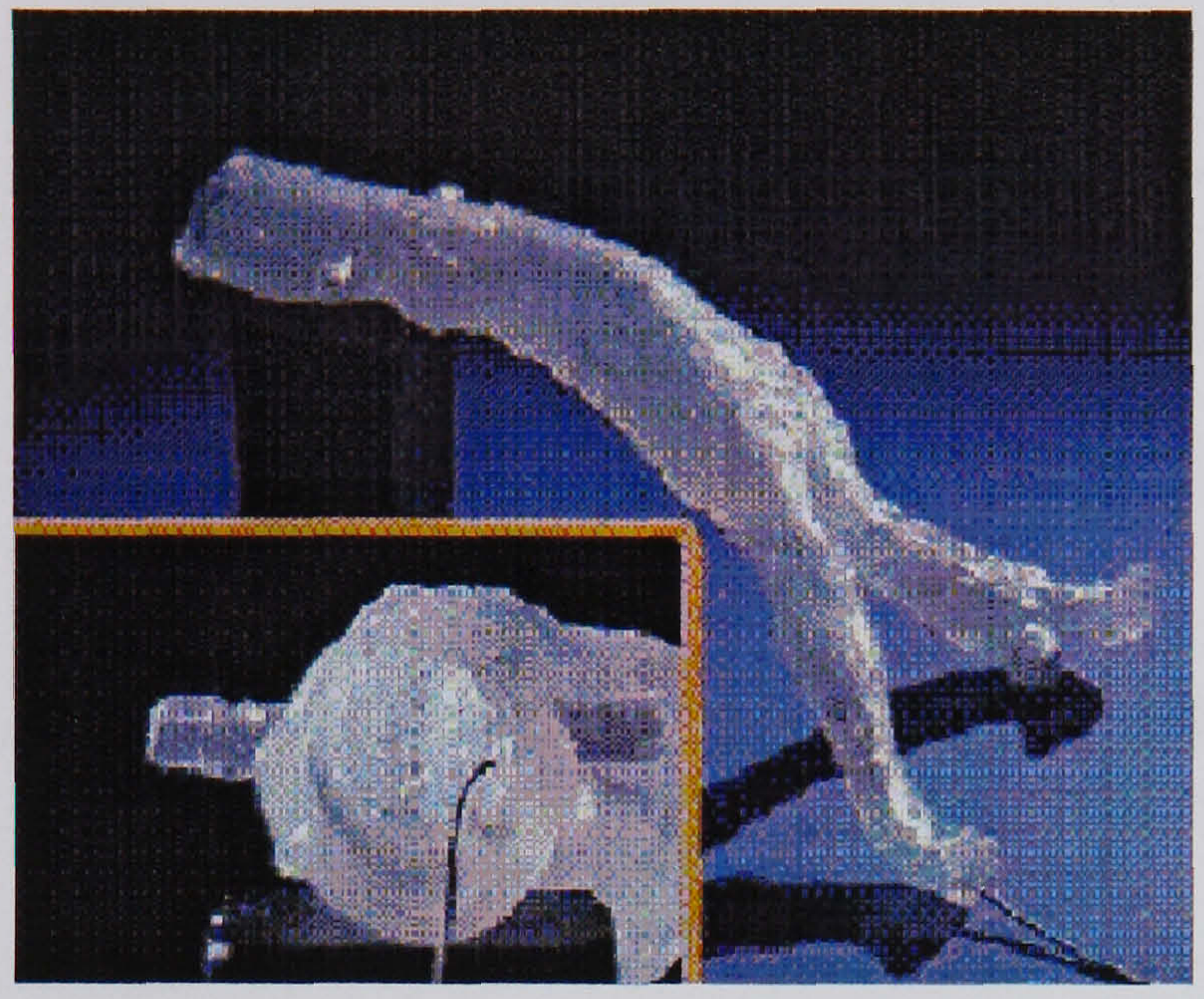
Abdominal Aortic Aneurysm Movies

These movies were captured at the University of Leeds, TV unit. They were all captured in a setting similar to our training simulator- i.e. showing a global view of the whole model as well as a local view. The model used is a silicon mockup of type02 model produced using rapid prototyping in the Mechanical engineering at the University of Leeds. The movies show manipulation of a J-shaped guide wire and a sidewinder catheter to access the renal artery- which is typical in a renal plasty procedure.



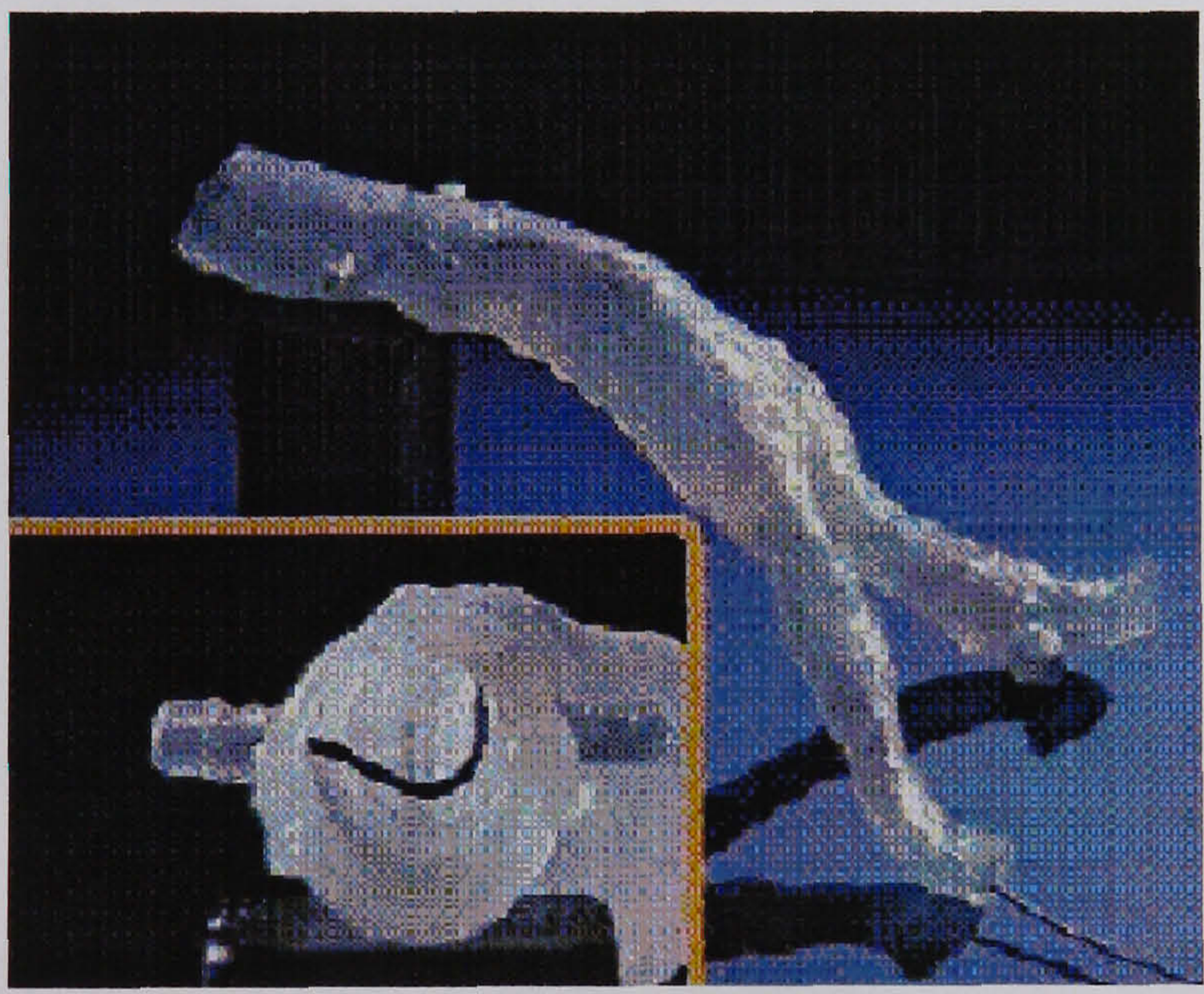
The first step is to insert the guide wire from the femoral artery to the aorta at a level above the renal artery.

- This movie shows an attempt to insert the guide wire that failed because the wire is buckled inside the aorta.
- This movie shows a successful insertion process.



The second step is to insert the catheter over the guide wire and to twist it until the tip of the catheter is facing the renal artery.

- This movie shows a catheter insertion process as well as possible difficulties when rotating the catheter. (This movie is annotated)



Once the catheter tip is facing the renal artery, push the guide wire inside. You will feel resistance when pushing the guide wire if the catheter is not facing the artery properly or if the catheter moves while pushing the guide wire. This movie shows these difficulties. (This movie is annotated)

Back to main page

Abdominal Aortic Aneurysm Links

Medical sites

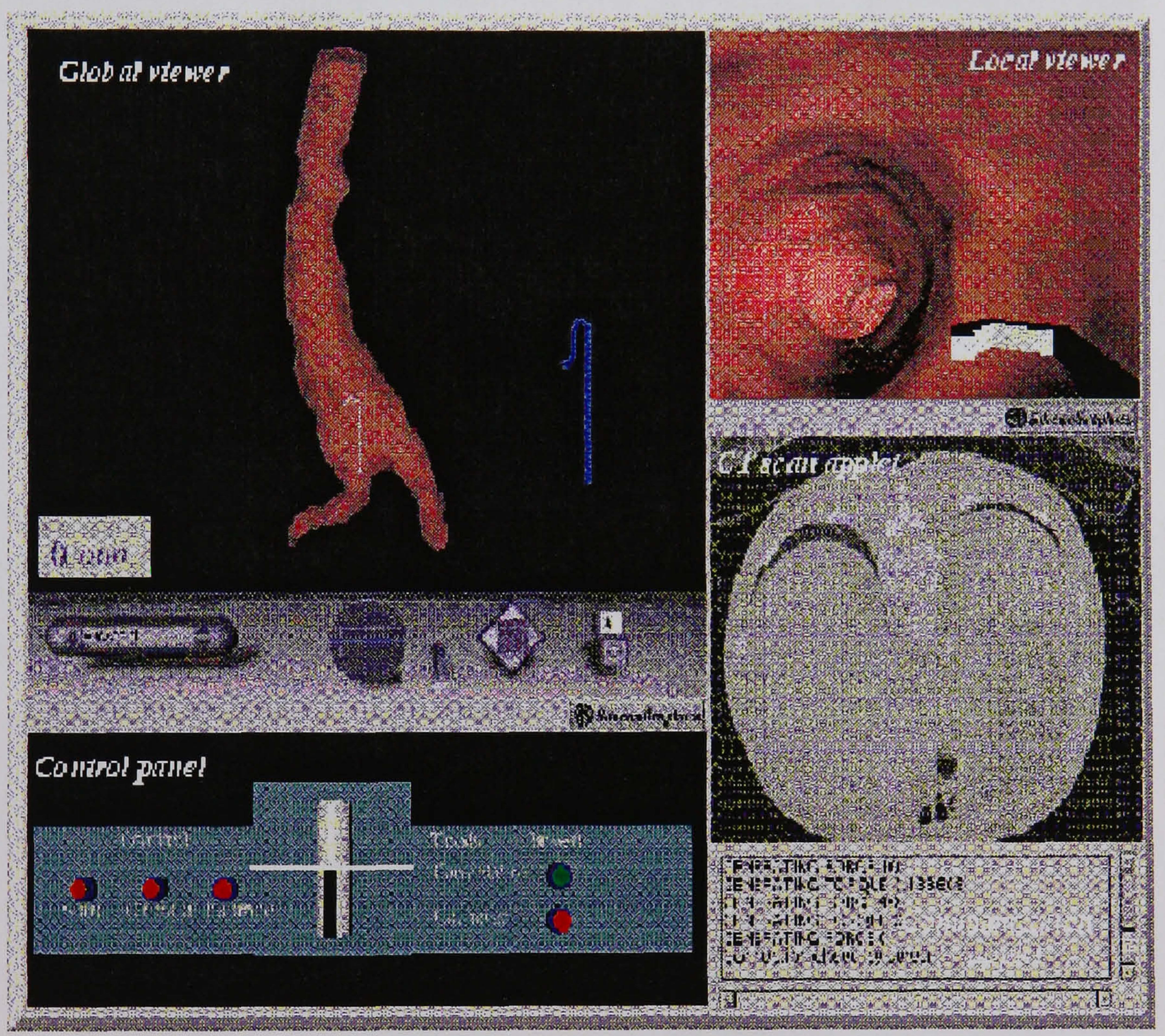
- Virtual Hospital: Abdominal Aortic Aneurysm.
- Mayo Clinic: Aortic aneurysm, Surgery can stop this silent danger.
- Abdominal Aortic Aneurysm: Intervention or Conservative Management?
- Ultrasound and Abdominal Aortic Aneurysm.
- Ruptured Abdominal Aortic Aneurysm.
- Medical College of Wisconsin: Collaborative Hypertext of Radiology
- Gloucestershire Vascular Group: Patient Information - Aortic Aneurysms.
- Health Answers: Abdominal aortic aneurysm.
- Imaginis.net: Vascular imaging.
- VIDA: Abdominal Aortic Aneurysm, images and volumes.
- Centre of Medical Imaging Research (Comir), University of Leeds: Aortic aneurysm training.

Other surgical simulators

- CieMed: da Vinici.
- HT medical: 'Dawson-Kaufman' Interventional Radiology simulator.
- University Hospital Utrecht: Fluoroscopy simulation.

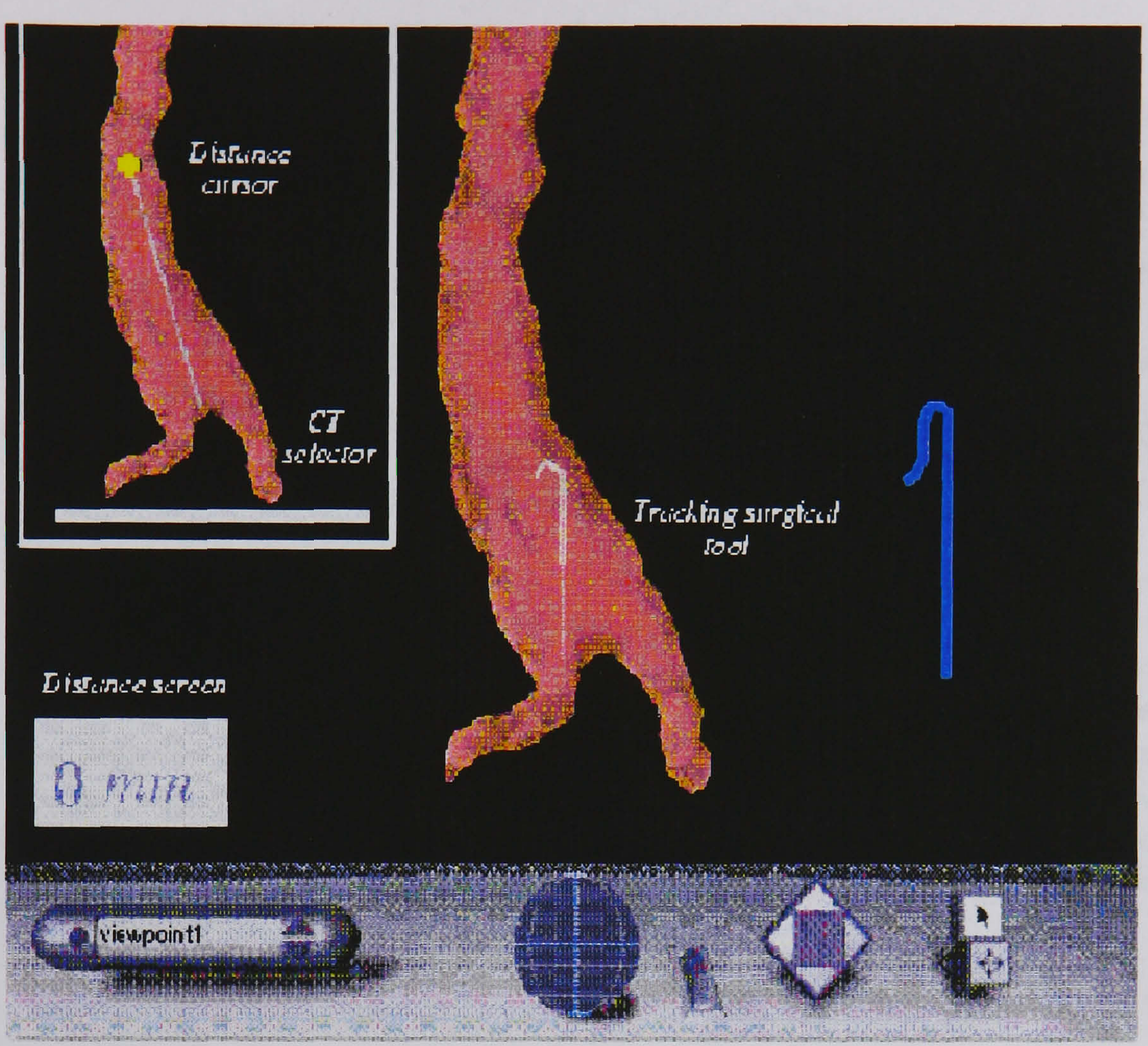
Back to main page

WebSTer, AAA Training Simulator

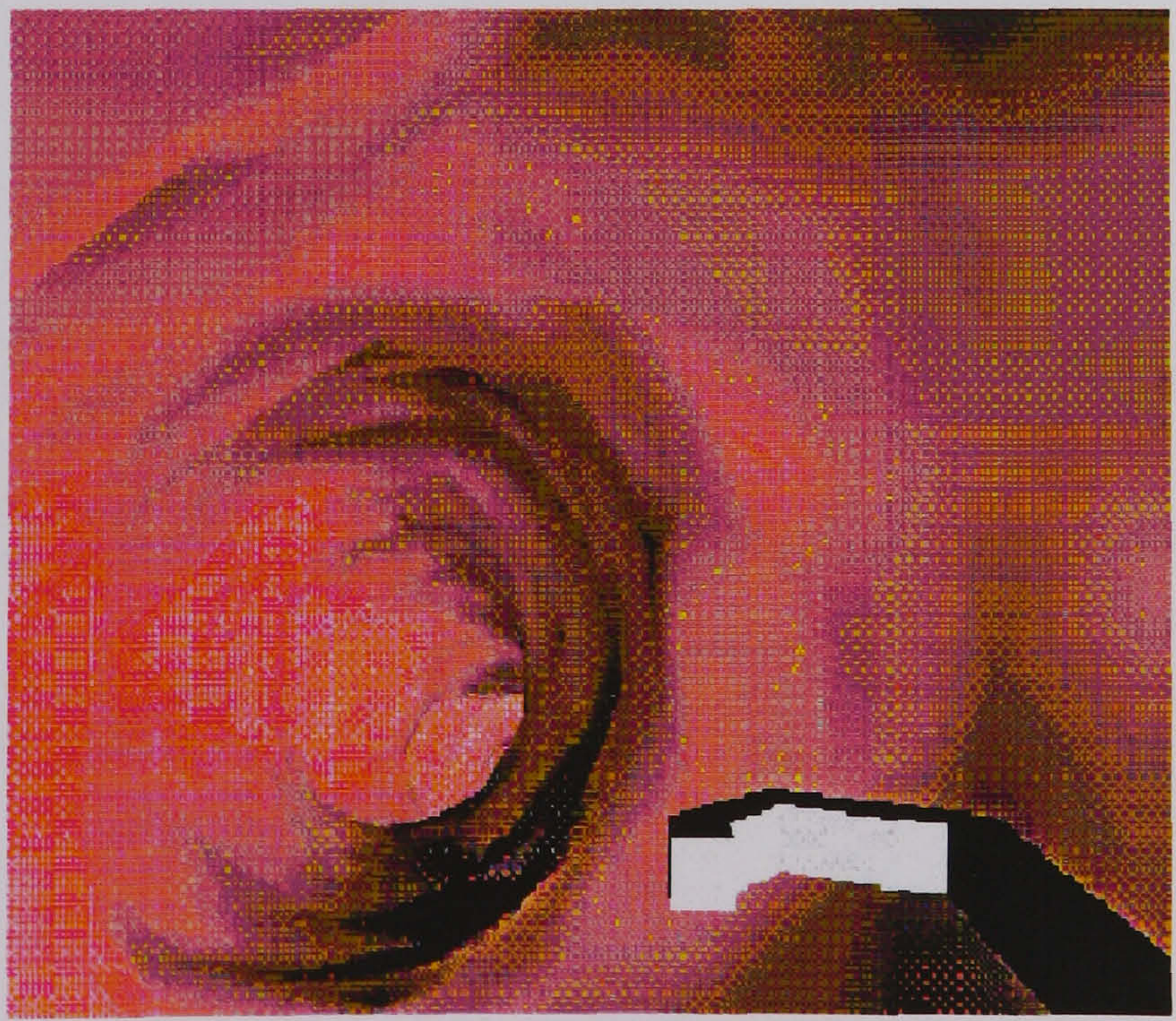


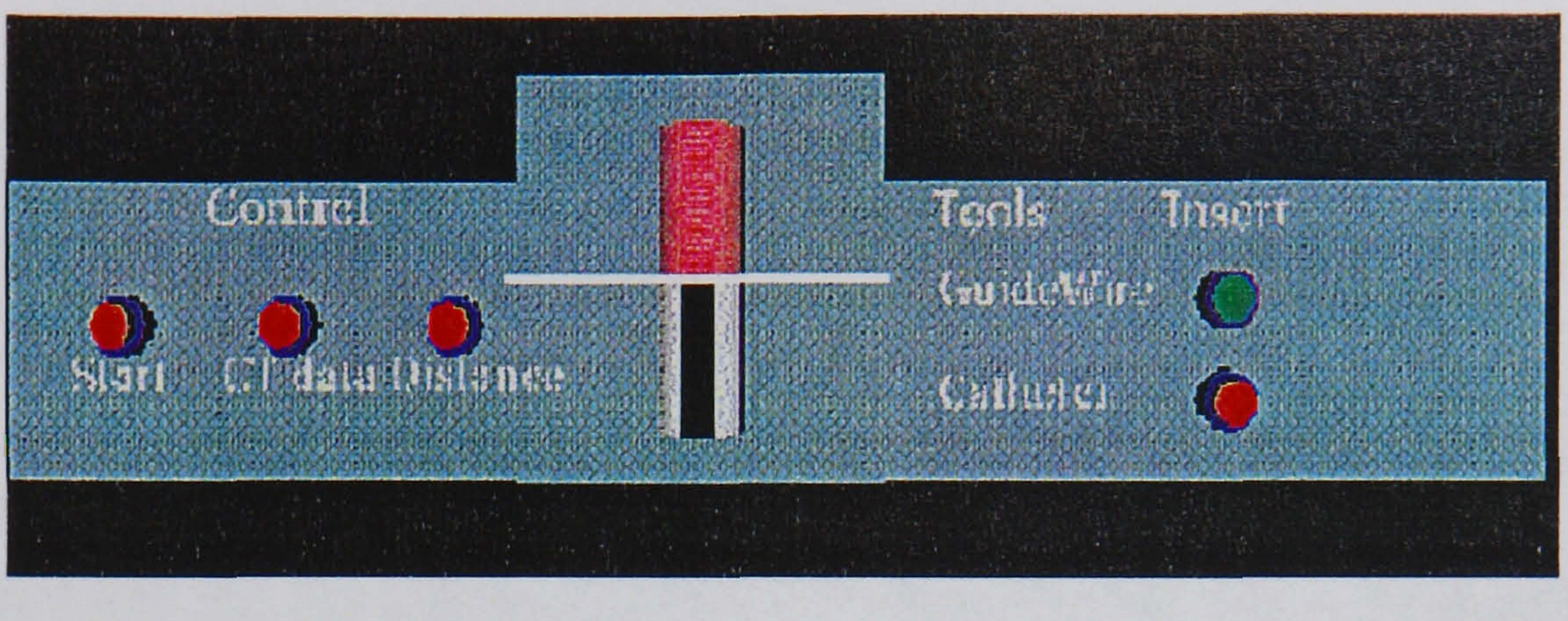
The AAA simulator consists of five windows:

- Global viewer.
- Local viewer.
- Control panel.
- CT scan applet.
- Communication applet.



The **global viewer** will show you an overall view of the aorta and the position of the surgical tool. In this viewer you can manipulate the distance tool to measure distances on the 3D model of the aorta. You can also scan the 3D model with the CT slicer tool to view the CT slice of the current position of the CT selector. The CT slice will be displayed on the **CT scan applet**.


The **local viewer** will show you an inside view of the vessel from the point of view of the current position of the tool.

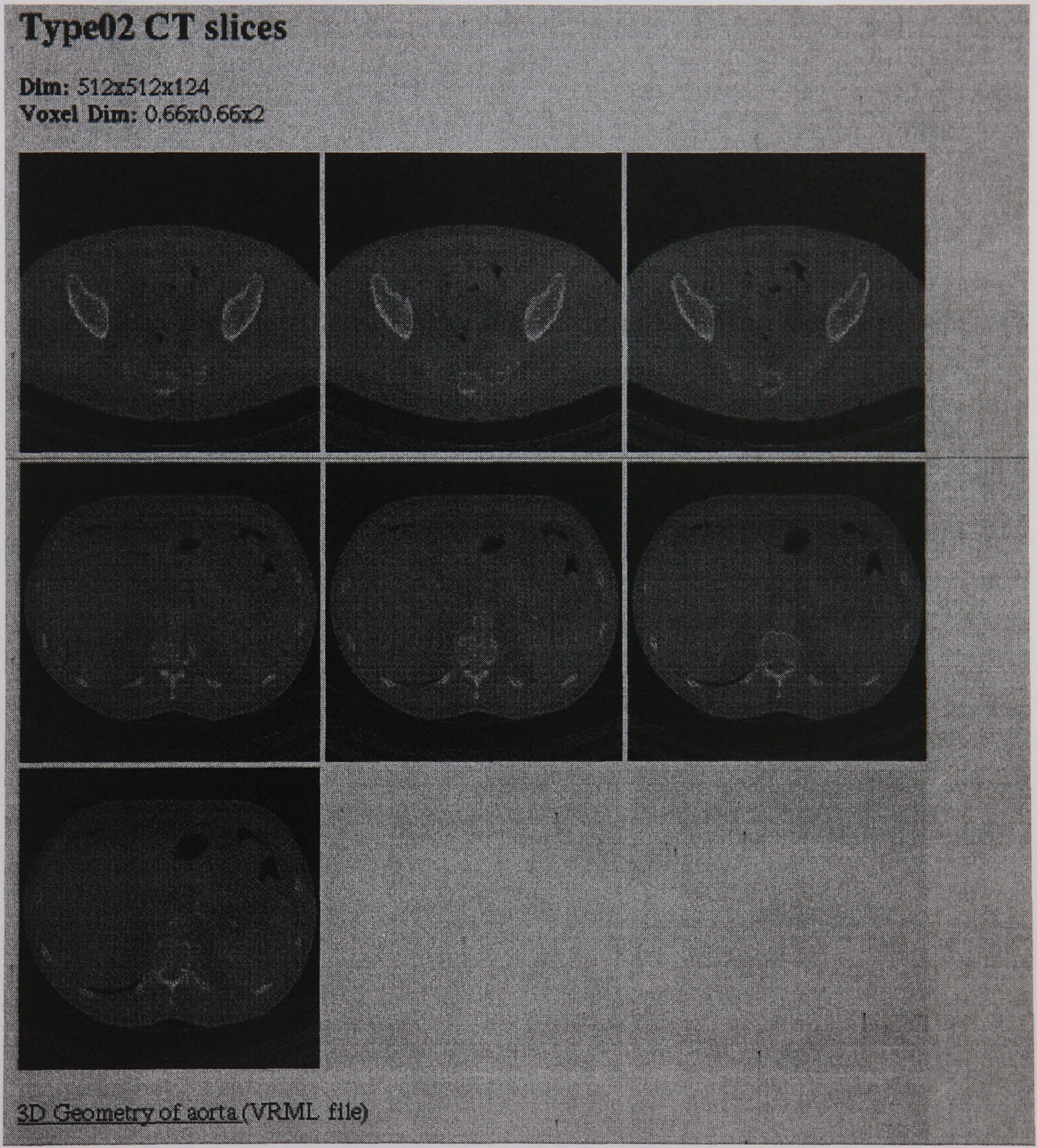


The **control panel** will allow you to manipulate the CT selector and distance tool. When the server is on, you can use this panel to activate and deactivate the surgical tools. The tool is inactive when its button is red. If you click on the tool button, the button will turn yellow and the tool is inserted in the proper position. Another click on the tool button will turn it to green to signify that the tool is now active and that forces and twists that you apply will be directed to this tool. You can apply forces and twists to the tools using the virtual scope- the cylinder in the middle of the panel. By clicking on the virtual scope and moving the mouse up and down, you apply push/ pull forces to the surgical tool. Meanwhile, if you move the mouse horizontally, you apply a clockwise/ counter clockwise twisting to the tool.

The **communication applet** displays messages to signify your actions and to report any errors.

We have a collection of AAA cases, click here: AAA models to choose the case you wish to work on.

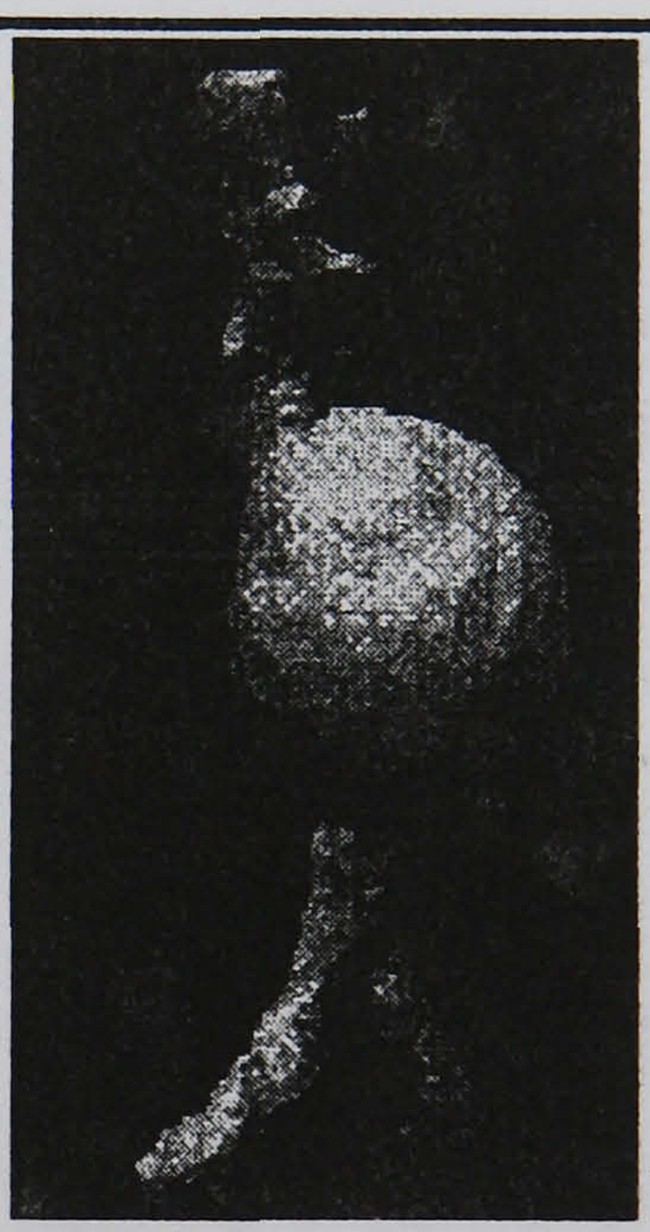
Back to main page

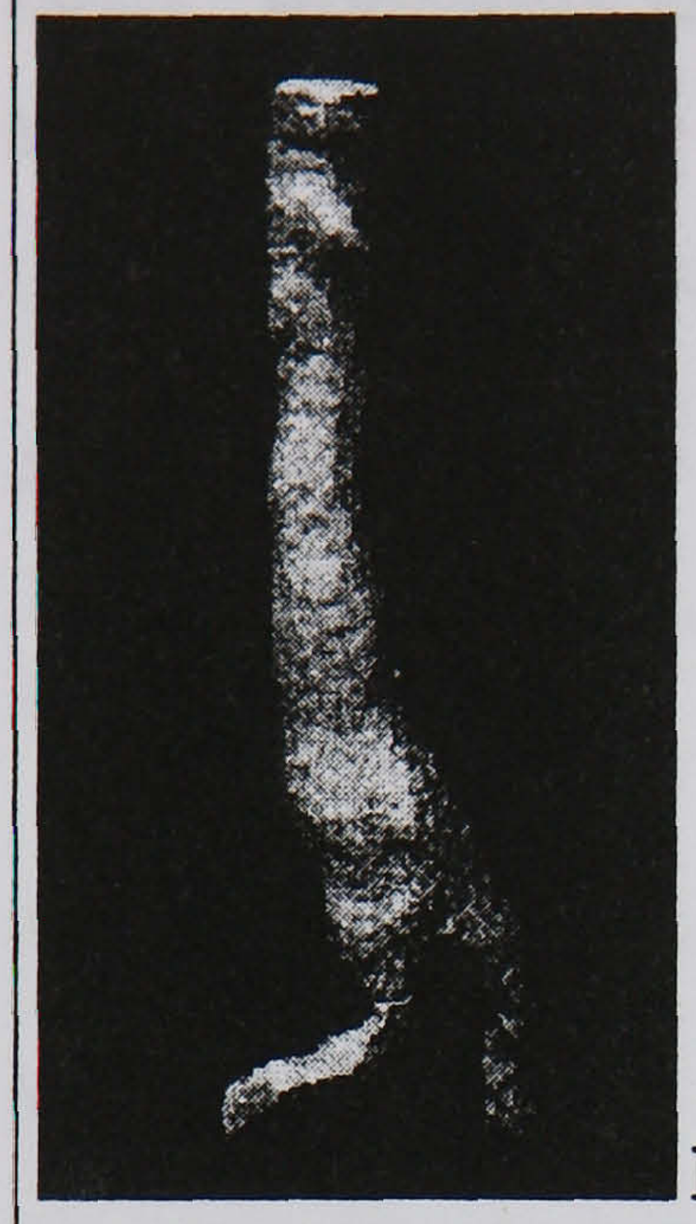

Type02 CT slices
Dim: 512x512x124
Voxel Dim: 0.66x0.66x2
3D Geometry of aorta (VRML file)

Abdominal Aortic Aneurysm Models

Following is the Abdominal Aortic Aneurysm (AAA) models that we use for our simulator. To try the AAA simulator :

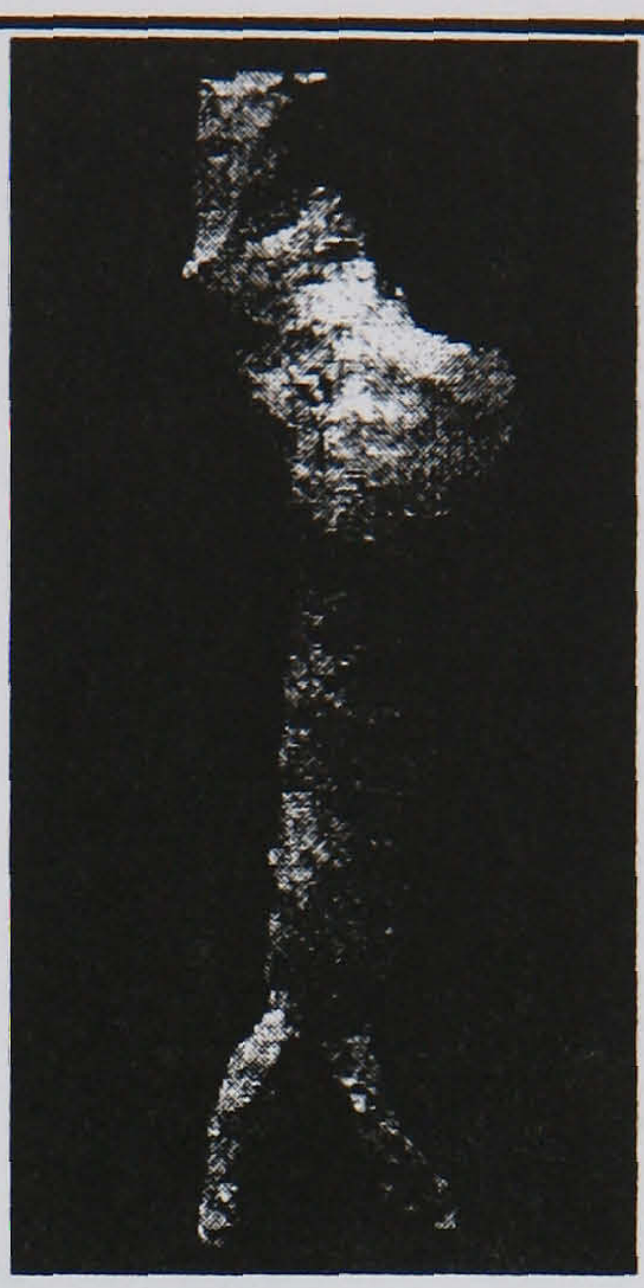
- Click on "Load procedure evaluation" link.
- Then, click the "start" link to load the simulator.

 **Model type_0.**	 **Model type_1.**
• Load procedure evaluation. • start	• Load procedure evaluation. • start



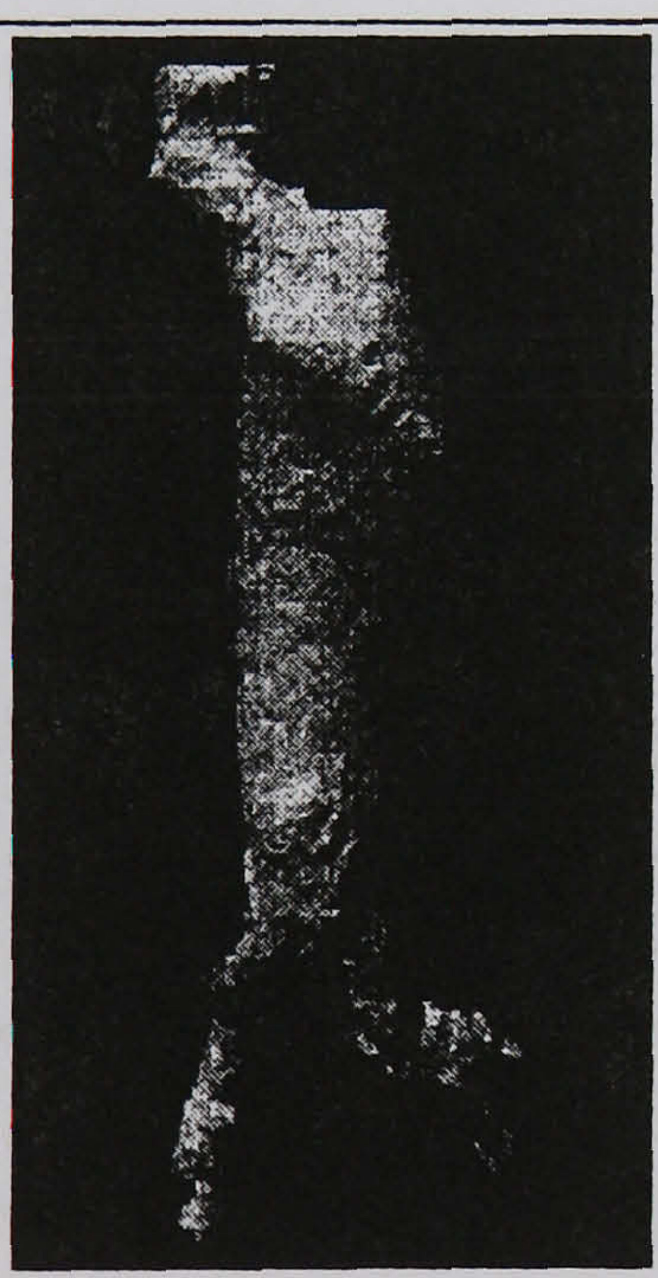
Model type_2.

- Load procedure evaluation.
- start



Model type_3.

- Load procedure evaluation.
- start



Model type_4.

- Load procedure evaluation.
- start

Model type_5.

- Load procedure evaluation.
- start

Back to main page

AAA - procedure evaluation

Measurement:

Use the distance tool to measure the following:

Aneurysm neck: [ ]

Aneurysm length: [ ]

Aneurysm to bifurcation distance: [ ]

Decision:

Is the case suitable for endovascular procedure: Yes ◇ No ◇

Comments:

Use this space to add comments about any expected difficulties in the procedure.

This evaluation belongs to : [ ] (please enter your name).

Model type number : [ ]

Send Clear

Appendix D

Evaluation process

- Page (162- 164) : Instruction sheet.
- Pages (165) : Evaluation sheet.

WebSTer Evaluation

Instructions sheet

The purpose of our research is to investigate the feasibility of building web-based surgical simulators for training purposes. The web supports multimedia technologies which can support all phases of training: verbal instructions, demonstration, practice and evaluation. We were able to develop a prototype of a web-based surgical training system (WebSTer) to train radiologists on interventional radiology procedures- Abdominal Aortic Aneurysm (AAA) is the case we study. An evaluation of this prototype will give an indication of how useful the web technology is for surgical training applications. In addition, it will identify limitations and future directions for such application. During this evaluation process, we will go through the different features of our system. After every stage, you will be asked to refer to the evaluation sheet and give a score to this feature and add your comments.

Verbal instructions & demonstration

We have prepared a set of web pages to give an example of how multi-media can be used to support the first two stages of a training- verbal instructions and demonstration. If you click on the link titled *AAA Endovascular Procedure*, description of the anatomy, surgical tools and the steps of the procedure are presented using a combination of text and images.

- Use the "*Back*" icon at the top left of the browser to get to the main page again.

Click the link titled *AAA Movies*, which presents a set of videos showing the manipulation of a guide wire and a catheter in a silicon mockup of the aorta. The required task was to advance the guide wire from the iliac artery to the renal branch, then to use a catheter to help in inserting the guide wire into the renal artery. The goal of such a video is to demonstrate the procedure of manipulating the surgical tools. In addition, it would give an idea of the difficulties expected. This type of video is useful in showing what is happening inside the aorta, as opposed to a video of the actual procedure in the theatre room which would show an external view of the procedure.

- Use the "*Back*" icon at the top left of the browser to get to the main page again.

In addition, the link titled *AAA Links* provides a number of links to other web sites interested in the same topic. It is useful for the trainee to be aware of other sources of information.

Note: The content of these web pages are not highly medical, but they exemplify what is possible.

Please refer to requirement (1) in the solution list and enter your grade and comments.

- Use the "*Back*" icon at the top left of the browser to get to the main page again.

- Click on the link titled *WebSTer: AAA Training Simulator*. This page will describe the different parts of our simulator. Please read the description to be able to use the simulator.

- At the bottom of the page, click the *AAA models* link to see the collection of AAA cases that we provide. We will be working with *model type_5*, because the renal arteries can be seen in this model. To work with this model, click the following two links:

- Click the link titled *Load procedure evaluation.* (A new window will appear on top of the first one, move the new window so you can view both together)
- Then, click the *start* link to launch our simulation environment.

Representational skill

If you click on the ***CT scan applet***- right middle window- another window will appear that shows a consecutive display of the CT slices of this case. This feature provides the clinical method of viewing patient's data. In order to train physicians to construct a mental 3D model out of the 2D slices, we provide the 3D construction of these slices in a link at the bottom of the page.

- You can close this window by clicking on the "X" icon at the top right of the window.

Another solution for this problem is to allow the physician to interactively scan the 3D model and view the corresponding CT slice. To try this solution, click the button titled *"CT data"* in the ***control panel***. A white bar will appear in the ***global viewer*** window. If you click and drag the bar along the 3D model, the ***CT scan applet*** will update the image to show the corresponding CT slice.

Please refer to requirement (2) in the solution list and enter your grade and comments.

- Click the *CT data* button in the ***control panel*** to switch it off.

Measurement procedure and decision making

The ***AAA- procedure evaluation*** page contains measurements that are necessary to decide the suitability of the case for an endovascular procedure. To perform these measurements, click on the *distance* button in the ***control panel*** of the simulator. A cross cursor will appear in the ***global viewer***. When the button is yellow, you can move the cursor around to put it in the initial position of the measurement. Click the *distance* button again and it will turn green. Now when you drag the cursor on the model, it will highlight the path swept and will display the distance of this path on the screen that appears on the left bottom corner of the ***global viewer***.

After taking the measurements of the aneurysm neck, aneurysm length and aneurysm to bifurcation distance. Decide whether this case is suitable for an endovascular procedure or not and add comments of any expected difficulties in this case. When you click the send button, this information will be written to a file and can be used later to evaluate the trainee.

Please refer to requirement (3) and (4) in the solution list and enter your grade and comments.

- Close the AAA procedure evaluation browser.

- Click the *distance* button in the ***control panel*** to switch it off.

Surgical tool simulation

To train the physicians to manipulate the surgical tools, we provide two tools- a guide wire and catheter. The guide wire is a J-shaped wire- drawn in white colour- in the ***global viewer*** window. We also provide a shepherd shaped catheter- drawn in blue colour. By clicking on the ***guide wire*** button on the ***control panel***, you insert the guide wire into the artery in a pre-specified location- and the button will turn yellow- called inserted mode. In order to manipulate the guide wire, click the ***guide wire*** button again, to turn it to green- called active mode. You can have more than one tool in an inserted mode, but only one tool is active.

When your guide wire is active, push the ***start*** button on the ***control panel*** to start the simulator. You can apply forces and rotation to the guide wire using the ***virtual scope***- on the middle of the ***control panel***. Click and drag the ***virtual scope*** upwards to apply a push force to the tool. You will see that the guide wire representation in the ***global viewer*** will move upwards as well. The tool will continue the motion as long as you are applying force. You can pull the surgical tool by dragging the ***virtual scope*** downwards- below the middle white line. By moving the ***virtual scope*** horizontally, you apply a twisting force on the surgical tool. The bottom part of the ***virtual scope*** is coloured to help you identify your rotation position- black is no rotation, red is clockwise rotation and blue is counter clockwise rotation. A white line is drawn to track the path of the tool in the ***global viewer***, while the ***local viewer*** window is updated to show the current internal view of the aorta.

During the motion of the guide wire, if the tool hits the walls of the aorta, it will bend and move away from the wall. The strength of collision is displayed as a red colour intensity on the top part of the ***virtual scope***. In addition, a sound will be displayed to draw your attention to the collision.

When you reach the renal artery, activate the catheter by clicking its button twice; it will replace the guide wire at its latest position and forces and rotations will be directed to it. Twist the catheter until its tip faces the renal artery entrance. Activate the guide wire- by clicking its button- and apply push force to advance the guide wire from the catheter tip into the renal artery.

Please refer to requirement (5) and (6) in the solution list and enter your grade and comments.

WebSTer Evaluation

Evaluation sheet

Requirements:

From our analysis of the endovascular procedure used to treat abdominal aortic aneurysm, we recognised that the following aspects are needed in a computer- based simulator for training:

1. Verbal instructions and demonstration of the procedural tasks.
2. Pictorial media to build the representational skill- construction of 3D mental structure out of 2D slices.
3. An environment to practise procedural tasks- such as the measurement task.
4. An environment with a wide range of anatomical models to practise the decision making process of performing the operation and positioning the stent.
5. Tactile media to build the force perception.
6. An environment to practise inserting the guide wire (GW) and the catheters to build the required eye/hand coordination.

Solutions:

For each of the previous aspects, we have provided a solution. Please grade our solutions according to the degree of their success in achieving their goals (Give a grade from 1-3 to indicated poor, satisfactory, good). Please give reasons or suggestions of improving each aspect.

Req.No.	Solution	Grade	Comments
1	HTML pages + movies.		
2	Consecutive display of CT slices + 3D model link.		
2	CT scan selection tool (interactive display)		
3	Distance tool in the simulator.		
4	Different models+ procedure evaluation form.		
5	Visual force feedback.		
6	GW & catheter behaviour simulation.		
6	GW & catheter manipulation tools.		
6	Global and local viewer in the simulator.		

- What do you see the potential of such web-based applications in the medical field?
- What single thing would most improve this work?
- General comments.

Bibliography

[1] *Abdominal aortic aneurysm.* Health answers, http://www.healthanswers.com/database/ami/converted/000162.html.

[2] *daVinci: Medical Simulator.* CieMed, Center for Information Enhanced Medicine, http://beast.cbmv.jhu.edu:8000/projects/ daVinci/daVinci.shtml.

[3] *External Authoring Interface.* EAI Working Group, http://www.vrml.org/ WorkingGroups/vrml-eai/.

[4] *Fakespace - 3D immersive visualization systems.* Fakespace Inc., http://www.sun.com/desktop/java3d/in_action/success1.html#fake.

[5] *Issues in Numerical Computing With Java.* Numerics Working Group of the Java Grande Forum, http://math.nist.gov/ javanumerics/issues.html.

[6] *Java 3D API Specification, Version 1.1.* http://java.sun.com/ products/ java-media/ 3D/ forDevelopers/ j3dguide/ j3dTOC.doc.html.

[7] *Java Native Interface.* http://java.sun.com/ docs/ books/ tutorial/ native1.1/TOC.html.

[8] *Java: Remote Method Invocation.* http://java.sun.com/ docs/books/ tutorial/rmi/TOC.html.

[9] *Java Tutorial: Object-Oriented Programming for the Internet.* http://java.sun.com/docs/ books/ tutorial/index.html.

[10] *Java Ultimate Math Package.* http://members.xoom.com/znerd/ java/jump/index.html.

[11] *Minimally Invasive Surgery Training.* Virtual Presence Manchester Royal Infirmary, http://www.ukvrforum.org.uk/CaseStudies/mist.htm.

[12] *NPAC- 3D Visible Human.* Northeast Parallel Architectures Center at Syracuse University, http://www.npac.syr.edu /projects/3Dvisiblehuman/VRML/ VRML2.0/MEDVIS/.

[13] *A Numerical Library for Java.* Visual Numerics, http://www.vni.com/ products/wpd/jnl/index.html.

[14] *RAPID, version 2.01.* University of North Carolina, http://www.cs.unc.edu/ geom/OBB/OBBT.html.

[15] *Virtual Reality Modeling Language Specification. Version 2.0.* http://vag.vrml.org/ VRML2.0/FINAL/ spec/index.html.

[16] *The W3 EMRS Project.* National Library of Medicine, http://www.emrs.org/publications/.

[17] *Advanced Surgical Planning- Interactive Research Environment.* Stanford University Cardiovascular Biomechanics Research Laboratory, http://www-med.stanford.edu/aspire/, 1998.

[18] *Development of an aortic aneurysm training model using rapid prototyping and data from spiral CT.* Centre of Medical Imaging Research (Comir), University of Leeds, http://www.scs.leeds.ac.uk/comir/research/aaa/aaa.htm, 1998.

[19] *Internet Bandwidth Management.* iBAND - Internet Bandwidth Management Summit, http://www.stardust.com/iband/backgrounder.pdf, 1998.

[20] *X-ray Imaging: Radiography and Fluoroscopy.* Imaginis Corporation., http://www.imaginis.net/xray/, 1999.

[21] A. Ali and R. Kikinis. *SPL Viz: Java3D Anatomy Browser.* http://www.sun.com/desktop/java3d/demos/entries/SPLViz/docs/index.html, 1998.

[22] A. Amini, R. Owen, and et. al. Non-rigid motion models for tracking the left ventricular wall. In A.C. Colchester and D.J. Hawkes, editors, *Lecture Notes in Computer Science: (Vol 511) Information Processing in Medical Imaging*, pages 343–57. Spring- Verlag, 1991.

[23] F. Arai, M. Ito, T. Fukuda, and et.al. Intelligent assistance for intravascular tele-surgery and experiments on virtual simulator. In *Proceeding of IEEE*

Annual International Symposium of Virtual Reality, pages 101–7, Triangle Park, March 1995.

[24] F. Arai, M. Tanimoto, and et. al. Distributed virtual environment for intravascular tele-surgery using multimedia telecommunication. In *Proceedings of the IEEE 1996 Virtual Reality Annual International Symposium*, pages 79–85, Los Alamitos,CA, 1996. IEEE Comput. Soc.

[25] M. Bajura, H. Fuchs, and R. Ohbuchi. Merging virtual objects with the real world: Seeing ultrasound imagery within the patient. In R. Taylor, S. Lavaliee, and et.al., editors, *Computer Integrated Surgery: Technology and Clinical applications*, chapter 15, pages 245–53. MIT Press, Lonodn, 1996.

[26] F. Balaguer and A. Mangili. Virtual environments. In D. Thalmann N. Magnenat-Thalmann, editor, *New Trends in Animation and Visualization*, chapter 6, pages 91–106. Wiley, 1991.

[27] D. Baraff. Analytical methods for dynamic simulation of non- penetrating rigid bodies. *Computer Graphics*, 23(3):223–231, 1989.

[28] D. Baraff. Non-penetrating rigid body simulation. In *Eurographics'93*, Barcelona, September 1993.

[29] D. Baraff. Rigid body simulation. In *SIGGRAPH'98: Course Notes*, pages D1–D48, Florida, USA, July 1998. ACM.

[30] D. Baraff and A. Witkin. Dynamic simulation of non-penetrating flexible bodies. *Computer Graphics*, 26(2):303–8, 1992.

[31] W. Barfield and T.A. Furness. *Virtual Environment and Advanced Interface Design.* Oxford University Press, New York, 1995.

[32] W. Barfield and C. Hendrix. Factors affecting presence and performance in virtual environments. In K. Morgan and R. Satava, editors, *Interactive Technology and New Paradigm for Healthacre*, pages 21–8. IOS Press and Ohmsha, 1995.

[33] W. Barfield, D. Zeltzer, T. Sheridan, and M. Slater. Presence and performance within virtual environments. In W. Barfield and T. Furness, editors, *Virtual Environments and Advanced Interface Design*, chapter 12, pages 473–513. Oxford University Press, 1995.

[34] V.L. Barker. Cathsim^{TM}. In J.D. Westwood and et.al., editors, *Medicine Meets Virtual Reality*, pages 36–37. IOS Press and Ohmsha, 1999.

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

[36] R. Barzel and A. Barr. A modelling system based on dynamic constraints. *Computer Graphics*, 22(4):179–88, 1988.

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

[38] E. Berglund and H. Eriksson. Distributed interactive simulation for group-distance exercises on the web. In P. Fishwick, D. Hill, and R. Smith, editors, *The International Conference on Web-Based Modelling and Simulation*, 30, pages 91–5, San Diego, California, January 1998. The Society for Computer Simulation International.

[39] C. Blohm and K. Hentschel. Liquid reality: Physics and locales. *http://www.vrmlsite.com/sep96/spotlight/lr/lr.html*, 1996.

[40] U. Blum, M. Langer, and et. al. Abdominal aortic aneurysms: Preliminary technical and clinical results with transfemoral placement of endovascular self-expanding stent-grafts. *Radiology*, 198(1):25–31, 1996.

[41] W. Broll and D. England. Bringing worlds together: Adding multi-user support to VRML. In *VRML'95*, San Diego, USA, December 1995.

[42] A. Bulpitt and N. Efford. An efficient 3D deformable model with a self-optimising mesh. *Image and Vision Computing*, 14:573–80, 1996.

[43] P. Cahoon and E. Grant. Telemedicine and shared multidimensional workspaces. *Computer Graphics*, pages 19–21, Feb 1996.

[44] G. Carraro, M. Cortes, J. Edmark, and J. Ensor. The Peloton bicycling simulator. In S. Spencer, editor, *Proceedings of VRML'98*, pages 63–70, Monterey, California, February 1998. ACM, Inc.

[45] J. Chadwick, D. Haumann, and R. Parent. Layered construction for deformable animated characteres. *Computer Graphics*, 23(3):243–52, 1989.

[46] P. Chapman and D. Wills. Towards a unified physical model for virtual environments. In R. Bowden, editor, *Proceedings of the 4th UK VRSIG conference*, pages 130–9, November 1997.

[47] P. Chapmen and D. Wills. Model analysis: A physical model for virtual environments? In *Proceedings of the 16th UK EG conference*, pages 175–182, March 1998.

[48] J. Cohen, M. Lin, and et.al. I-COLLIDE: an interactive and exact collision detection system for large- scale environments. In *Proceedings of ACM Int. 3D Graphics Conference*, pages 189–196, 1995.

[49] A. Colchester, J. Zhao, and et.al. Development and preliminary evaluation of VISLAN, a surgical planning and guidance system using intra-operative video imaging. *Medical Image Analysis*, 1(1):73–90, 1996.

[50] J. Coleman, A. Goettsch, and et.al. *TeleInViVo*^{TM}: Towards collaborative volume visualization environments. *Computers and Graphics*, 20(6):801–811, 1996.

[51] S. Cotin, H. Delingette, and et. al. Geometric and physical representation for a simulation for haptic surgery. In H. Sieburg, S. Weghorst, and K. Morgan, editors, *Health Care in the Information Age*, chapter 17, pages 139–51. IOS Press and Ohmsha, 1996.

[52] S. Cover, N. Ezquerra, and J. O'Brien. Interactively deformable models for surgery simulation. *IEEE Computer graphics and applications*, 13(6):68–75, 1993.

[53] S. Delp, P. Loan, C. Basdogan, and et.al. Surgical simulation: An emerging technology for training in emergency medicine. *Presence*, 6(2):147–59, April 1997.

[54] M. Desbrun and M. Gascuel. Highly deformable material for animation and collision processing. In *Fifth Eurographics Workshop on Animation and Simulation*, Oslo, Norway, September 1994.

[55] M.P. do Carmo. *Differential geometry of curves and surfaces.* Prentice-Hall, London, 1976.

[56] W. Doyle. Interactive image-directed epilepsy surgery: Rudimentary virtual reality in neurosurgery. In K. Morgan and R. Satava, editors, *Interactive Technology and New Paradigm for Healthacre*, pages 91–100. IOS Press and Ohmsha, 1995.

[57] C.G. Drury, B. Paramore, and et.al. Task analysis. In G. Salvendy, editor, *Handbook of Human Factors*, chapter 3.4, pages 370–401. Wiley-Interscience Publication, 1987.

[58] A.C. Dumay and G.J. Jense. Endoscopic surgery simulation in a virtual environment. *Computers in Biology and Medicine*, 25(2):139–48, 1995.

[59] R. Durikovic, K. Kaneda, and H. Yamashita. Reconstructing a 3-D structure with multiple deformable solid primitives. *Computer and Graphics*, 21(5):611–24, 1997.

[60] R.A. Earnshaw, M. Gigante, and H. Jones. *Virtual Reality systems.* Academic, London, 1993.

[61] B. Eberhardt, A. Weber, and W. Strasser. A fast, flexible, particle-system model for cloth draping. *IEEE Computer graphics and Applications*, pages 52–9, September, 1996.

[62] E. Martin (ed.). *Concise Medical Dictionary.* Oxford University Press, 1980.

[63] J. Edwards and C. Hand. Maps: Movement and planning support for navigation in an immersive VRML browser. In *Proceedings of The Second Symposium on the Virtual Reality Modeling Language (VRML'97)*, pages 65–73, Monterey, California, February 1997. ACM.

[64] M. Elchanani and T. Sobh. World Wide Web operated robot arm. In P. Fishwick, D. Hill, and R. Smith, editors, *The International Conference on Web-Based Modelling and Simulation*, 30, pages 77–84, San Diego, California, January 1998. The Society for Computer Simulation International.

[65] I. Essa, S. Sclaroff, and A. Pentland. A unified approach for physical and geometric modelling for graphics and animation. *Computer Graphics Forum*, 2(3), 1992.

[66] J.D. Foley, A. van Dam, S.K. Feiner, and J.F. Hughes. *Computer Graphics: Principles and practice.* Addison-Wesley Publishing Company, New York, second edition, 1996.

[67] B. Frohlich, G. Grunst, and et.al. The responsive workbench: A virtual working environment for physicians. *Computers in Biology and Medicine*, 25(2):301-8, 1995.

[68] Y.C. Fung. *Biomechanics: Mechanical Properties of Living Tissues*. Springer-Verlag, New York, 1993.

[69] M. Gascuel and M. Desbrun. Animation of deformable models using implicit surfaces. *IEEE Transactions on Visualization and Computer Graphics*, 3(1), 1997.

[70] B. Geiger and R. Kikinis. Simulation of endoscopy. In Nicholas Ayache, editor, *International Conference on Computer Vision, Virtual Reality, and Robotics in Medicine, CVR Med '95*, pages 277–81, Nice, France, 1995. Springer-Verlag.

[71] J.M. Gere and S. Timoshenko. *Mechanics of materials*. Chapman and Hall, London, third edition, 1991.

[72] S. Gibson, C. Fyock, and et.al. Volumetric object modelling for surgical simulation. *Medical Image Analysis*, 2(2):121–32, 1998.

[73] T. Goodman and B. Ong. Shape preserving interpolation by space curves. *Computer Aided Geometric Design*, 15:1–17, 1997.

[74] J.P. Gourret, N.M. Thalmann, and D. Thalmann. Simulation of object and human skin deformations in a grasping task. *Computer graphics*, 23(3):21–30, 1989.

[75] Z. Guan, J. Ling, and et.al. Study and application of physics-based deformable curve and surfaces. *Computer and Graphics*, 21(3):305–13, 1997.

[76] U. Gudukbay and B. Ozguc. An animation system for rigid and deformable models. *Computer and Graphics*, 17(1):71–77, 1993.

[77] U. Gudukbay and B. Ozguc. Animation of deformable models. *Computer Aided Design*, 26(12):868–75, 1994.

[78] U. Gudukbay and B. Ozguc. A spring force formulation for elasticity deformable models. *Computer and Graphics*, 21(3):335–46, 1997.

[79] A. Gunkel, W. Freysinger, and et.al. Application of the ARTMA image-guided navigation system to endonasal sinus surgery. In *Computer Assisted Radiology: 9th International Symposium and Exhibition*, Berlin, June 1995.

[80] C. Hand. Other faces of virtual reality. In *Proceedings of the East-West International Conference on Multimedia, Hypermedia and Virtual Reality*, pages 69–74, Moscow, Russia, September 1994.

[81] C. Hand. A survey of 3D interaction techniques. *Computer Graphics forum*, 16(5):269–81, 1997.

[82] M. Harreld, D. Valentino, and W. Karplus. The virtual aneurysm: Virtual reality in endovascular therapy. In S. Weghorst H. Sieburg and K. Morgan, editors, *Health Care in the Information Age*, pages 12–20. IOS Press and Ohmsha, 1996.

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

[84] C. Hendrix and W. Barfield. Presence within virtual environments as a function of visual display parameters. *Presence*, 5(3):274–89, 1996.

[85] G.A. Higgins, G.L. Merrill, and et.al. New simulation technologies for surgical training and certification: Current status and future projects. *Presence*, 6(2):160–72, April 1997.

[86] H.M. Hoffman. Virtual reality meets medical education. In K. Morgan, R. Satava, and H. Sieburg, editors, *Interactive Technology and the New Paradigm for Healthcare*, chapter 22, pages 130–6. IOS Press and Ohmsha, 1995.

[87] D. Holding. Concepts of training. In G. Salvendy, editor, *Handbook of Human Factors*, chapter 8.2, pages 939–62. Wiley-Interscience Publication, 1987.

[88] D. Hon. Medical reality and virtual reality. In H. Sieburg, S. Weghorst, and K. Morgan, editors, *Health Care in the Information Age*, chapter 41, pages 327–41. IOS Press and Ohmsha, 1996.

[89] L. Hong, S. Muraki, and et.al. Virtual voyage: Interactive navigation in the human colon. In *SIGGRAPH'97*, pages 27–34, Los Angeles, April 8 1997.

[90] R. Hubbold, A. Murta, A. West, and T. Howard. Design issues for virtual reality systems. In *First Eurographics Workshop on Virtual Environments*, Barcelona, September 1993.

[91] A. Hubler and A. Assad. CyberProf: An intelligent human computer interface for asynchronous wide area treaning and teaching. In *Fourth International World Wide Web Conference : the Web revolution*, pages 231–8, Boston, Massachusetts, December 1995. International World Wide Web Conference, O'Reilly and Assoc.

[92] T. Hudson, M. Lin, and et.al. V-COLLIDE: Accelerated collision detection for VRML. In *VRML'97*, http://www.cs.unc.edu/ dm/collision.html, 1997.

[93] I.W. Hunter, L.A. Jones, and et. al. Opthalmic microsurgical robot and associated virtual environment. *Computers in Biology and Medicine*, 25(2):173–82, 1995.

[94] G. Iazeolla and A. D'Ambrogio. A web-based environment for the reuse of simulation models. In P. Fishwick, D. Hill, and R. Smith, editors, *The International Conference on Web-Based Modelling and Simulation*, 30, pages 37–42, San Diego, California, January 1998. The Society for Computer Simulation International.

[95] P. Jackson. Application of virtual reality in training simulation. In J. Gray, K. Warwick, and D. Roberts, editors, *Virtual Reality in Engineering*, pages 121–136. 1993.

[96] N. John, N. Phillips, R. Vawda, and J. Perrin. A VRML simulator for ventricular catheterisation. In *Eurographics UK'99*, Cambridge, UK, April 1999.

[97] R. Johnston, S. Bhoyrul, and et. al. Assessing a virtual reality surgical skills simulator. In H. Sieburg, S. Weghorst, and K. Morgan, editors, *Health Care in the Information Age*, chapter 66, pages 608–17. IOS Press and Ohmsha, 1996.

[98] E. Jordan and I. Kohane. *A Web-based Interface to Childhood Immunization Registries.* AMIA Electronic Poster, http://www.emrs.org/publications/, 1996.

[99] M. Riemer K. Hohne, M. Bomans and et.al. A 3D anatomical atlas based on a volume model. *IEEE Computer Graphics and Applications*, 12(4):72–78, 1992.

[100] R. Kalawsky. *The Science of Virtual Reality and Virtual Environments: A technical, scientific and engineering reference on visual environments.* Addison-Wesley, Wokingham, 1993.

[101] V. Kamat. A survey of techniques for simulation of dynamic collision detection and response. *Computer and Graphics*, 17(4):379–85, 1993.

[102] A. Kancherla, J. Rolland, and et.al. A novel virtual reality tool for teaching dynamic 3D anatomy. In Nicholas Ayache, editor, *International Conference on Computer Vision, Virtual Reality, and Robotics in Medicine, CVR Med '95*, pages 277–81, Nice, France, 1995. Springer-Verlag.

[103] R. Kenyon and M. Afenya. Training in virtual and real environments. *Annals of Biomedical Engineering*, 23:445–55, 1995.

[104] K. Khodabandehloo, P. Brett, and R. Buckingham. Special purpose actuators and architectures for surgery robots. In R. Taylor, S. Lavaliee, and et.al., editors, *Computer Integrated Surgery: Technology and Clinical applications*, chapter 17, pages 263–75. MIT Press, Lonodn, 1996.

[105] J. Knight. Manual control and tracking. In G. Salvendy, editor, *Handbook of Human Factors*, pages 182–218. Wiley-Interscience Publication, 1987.

[106] K. Kobayashi. Human, computers and communication. In G. Salvendy, editor, *Handbook of Human Factors*, chapter 12.7, pages 1790–1809. Wiley-Interscience Publication, 1987.

[107] J. Kozak, P. Hancock, E. Arthur, and S. Chrysler. Transfer of training from virtual reality. *Ergonomics*, 36(7):777–84, 1993.

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

[109] A. Lasko-Harvill, C. Blanchard, and et.al. A fully immersive cholecystectomy simulation. In K. Morgan, R. Satava, and H. Sieburg, editors, *Interactive Technology and the New Paradigm for Healthcare*, chapter 31, pages 182–6. IOS Press and Ohmsha, 1995.

[110] G. Leach, G. Al-Qaimari, M. Grieve, and et.al. Elements of a three dimensional graphical user interface. In S. Howard, J. Hammond, and G. Lindgaard, editors, *Human-Computer Interaction: INTERACT'97*, pages 69–76. Chapman and Hall, 1997.

[111] Y. Lee, D. Terzopoulos, and K. Waters. Realistic modeling for facial animation. In *Computer Graphics Proceedings*, pages 55–61, Los Angeles, August 1995.

[112] R. Lehman. *Computer Simulation and Modelling: An Introduction.* Lawrence Erlbaum Associates, New Jersey, 1977.

[113] I.P. Logan, J.W. Ward, and et.al. Arthroscopic training in a virtual environment: is it a reality? In *Proceedings of Med Tec'97, Medical Technology Conference and Exposition*, 1997. http://www.enc.hull.ac.uk/ cspipl/.

[114] I.P. Logan, D. Wills, and N.J. Avis. Deformable objects in virtual environements. In *Proceedings of the Second UK VR-SIG Conference*, Reading, United Kingdom, December 1994.

[115] I.P. Logan, D. Wills, and et. al. Virtual environment knee arthroscopy training system. *Society for Computer Simulation, Simulation Series*, 28(4):17–22, 1996.

[116] W. Lorensen, F. Jolesz, and R. Kikinis. The exploration of cross-sectional data with a virtual endoscope. In K. Morgan, R. Satava, and H. Sieburg, editors, *Interactive Technology and the New Paradigm for Healthcare*, chapter 36, pages 221–30. IOS Press and Ohmsha, 1995.

[117] H. Mann and L. Waldmann. Web-based modelling and simulation of multidisciplinary engineering systems. In P. Fishwick, D. Hill, and R. Smith, editors, *The International Conference on Web-Based Modelling and Simulation*, 30, pages 183–8, San Diego, California, January 1998. The Society for Computer Simulation International.

[118] Y. Masutani, K. Masamune, and et.al. Computer aided surgery (CAS) system for stereotactic neurosurgery. In Nicholas Ayache, editor, *International Conference on Computer Vision, Virtual Reality, and Robotics in Medicine, CVR Med '95*, pages 247–51, Nice, France, 1995. Springer-Verlag.

[119] T. McInerney and D. Terzopoulos. Deformable models in medical image analysis: A survey. *Medical Image Analysis*, 1(2), 1996.

[120] D. Meglan, R. Raju, and et. al. The Teleos virtual environment toolkit for simulation-based surgical education. In S. Weghorst H. Sieburg and K. Mor-

gan, editors, *Health Care in the Information Age*, pages 346–51. IOS Press and Ohmsha, 1996.

[121] J.R. Merril, G.L. Merril, and et.al. Photorealistic interactive three-dimensional graphics in surgical simulation. In K. Morgan and R. Satava, editors, *Interactive Technology and New Paradigm for Healthacre*, pages 244–52. IOS Press and Ohmsha, 1995.

[122] G. Miller. The motion of snakes and worms. *Computer Graphics*, 22(4):169–73, 1988.

[123] M. Mine, F. Brooks, and C. Sequin. Moving objects in space: Exploiting proprioception in virtual environment interaction. *Computer Graphics Proceedings, SIGGRAPH'97*, pages 19–26, Aug 1997.

[124] S. Miyazaki, M. Ishiguro, and et. al. A study of virtual manipulation of elastic objects. *Computer Graphics: Developments in Virtual Environments*, pages 381–91, 1995.

[125] M. Moore and J. Wilhelms. Collision detection and response for computer animation. *Computer graphics*, 22(4):289–98, 1988.

[126] K. Mori, J. Hasegawa, and et.al. Automated extraction and visualization of bronchus from 3D CT images of lung. In Nicholas Ayache, editor, *International Conference on Computer Vision, Virtual Reality, and Robotics in Medicine, CVR Med '95*, pages 542–48, Nice, France, 1995. Springer-Verlag.

[127] R. Mosges and S. Lavallee. Multimodal information for computer-integrated surgery. In R. Taylor, S. Lavaliee, and et.al., editors, *Computer Integrated Surgery: Technology and Clinical applications*, chapter 1, pages 5–19. MIT Press, Lonodn, 1996.

[128] W. Nowacki. *Dynamics of Elastic Systems.* Chapman and Hall Ltd., London, 1963.

[129] R. Orfali, D. Harkey, and J. Edwards. *Instant CORBA.* John Willey and Sons, Inc., New York, 1997.

[130] D. Ota, B. Loftin, and et.al. Virtual reality in surgical education. *Computers in Biology and Medicine*, 25(2):127–138, 1995.

[131] R. O'Toole, B. Jaramaz, and et. al. Biomechanics for preoperative planning and surgical simulators in orthopaedics. *Computers in Biology and Medicine*, 25(2):183–91, 1995.

[132] H. Oyama. System integration of VR simulated surgical support system. In M. Goebel, P. Slavik, and J.J. van Wijk, editors, *Virtual Environments and Scientific Visualisation '96*, pages 131–6. Springer Computer Science, 1996.

[133] H. Oyama, K. Nomura, and et.al. Surgical simulation support system. In K. Morgan and R. Satava, editors, *Interactive Technology and New Paradigm for Healthacre*, pages 439–44. IOS Press and Ohmsha, 1995.

[134] J. Peifer, W. Curtis, and M. Sinclair. Applied virtual reality for simulation for endoscopic retrograde cholangio-pancreatography (ERCP). In S. Weghorst H. Sieburg and K. Morgan, editors, *Health Care in the Information Age*, pages 36–42. IOS Press and Ohmsha, 1996.

[135] J. Peifer and R. Haleblian. Virtaul environment for eye surgery simulation. In *Medicine Meets Virtual Reality II*, pages 166–9, San Diego, January 1994.

[136] A. Pentland and J. Williams. Good vibrations: Modal dynamics for graphics and animation. *Computer Graphics*, 23(3):215–22, 1989.

[137] J. Platt and A. Barr. Constraint methods for flexible models. *Computer Graphics*, 22(4):279–88, 1988.

[138] K. Ramaswamy and W. Higgins. Endoscopic explorartion and measurement in 3D radiological images. In *SPIE conference of Medical Imaging*, Newport Beach, February 1996. http://cobb.ece.psu.edu/publications/ summary_page/summary_page.html.

[139] J.W. Regian. Virtual reality for training: Evaluating transfer. In R.J. Seidel and P.R. Chatelier, editors, *Virtual reality, training's future? : perspectives on virtual reality and related emerging technologies*, chapter 3, pages 31–40. Plenum Press, 1997.

[140] M. Rhodes. Computer graphics and medicine: A complex partnership. *IEEE Computer Graphics and Applications*, 17(1):22–28, 1997.

[141] R. Robb. VR assisted surgery planning: Using patient specific anatomic models. *IEEE Engineering in Medicine and Biology*, 15(2):60–69, 1996.

[142] R. Robb and B. Cameron. Virtual reality assisted surgery program. In K. Morgan and R. Satava, editors, *Interactive Technology and New Paradigm for Healthacre*, pages 309–21. IOS Press and Ohmsha, 1995.

[143] R. Robb and D. Hanson. The ANALYZE software system for visualization and analysis in surgery simulation. In R. Taylor, S. Lavaliee, and et.al., editors, *Computer Integrated Surgery: Technology and Clinical applications*, chapter 10, pages 175–189. MIT Press, Lonodn, 1996.

[144] J. Robinson, D. Hunter, and et. al. A new torque guide wire. *Radiology*, 165(2), 1987.

[145] B. Roehl. *Some Thoughts on Behaviour in VR Systems.* http://sunee.uwaterloo.ca/ broehl/behav.html.

[146] K. Rold and D. Smith. Guide wire with a J-shaped tip and a tapered movable core. *Radiology*, 149(1):317, 1983.

[147] S.M. Roth, M.H. Gross, and et.al. A bernstein-bezier based approach to soft tissue simulation. *EUROGRAPHICS'98*, 17(3):285–94, 1998.

[148] G. Rubin, C. Beaulieu, and et.al. Perspective volume rendering of CT and MR images: Applications for endoscopic imaging. *Radiology*, 199(2):321–30, 1996.

[149] H. Sarjoughian and B. Zeigler. DEVSJAVA: Basis for a DEVS-based collaborative M&S environment. In P. Fishwick, D. Hill, and R. Smith, editors, *The International Conference on Web-Based Modelling and Simulation*, 30, pages 29–36, San Diego, California, January 1998. The Society for Computer Simulation International.

[150] R. Satava. Medicine 2001: The king is dead. In K. Morgan and R. Satava, editors, *Interactive Technology and New Paradigm for Healthacre*, pages 334–39. IOS Press and Ohmsha, 1995.

[151] R. Satava. Medical virtual reality: The current status of the future. In S. Weghorst H. Sieburg and K. Morgan, editors, *Health Care in the Information Age*, pages 100–6. IOS Press and Ohmsha, 1996.

[152] R. Satava and R. Robb. Virtual endoscopy: Application of 3D visualization to medical diagnosis. *Presence*, 6(2):179–97, April 1997.

[153] J. Schroder. The mechanical properties of guide wires, Part I: Stiffness and torsional strength. *Cardiovascular and Interventional Radiology*, 16:43– 6, 1993.

[154] J. Schroder. The mechanical properties of guide wires, part III: Sliding friction. *Cardiovascular and Interventional Radiology*, 16:93–7, 1993.

[155] T. Seamster, R. Redding, and G. Kaempf. *Applied Cognitive Task Analysis in Aviation.* Avebury Ashgate Publishing Ltd., England, 1997.

[156] C. Shen. Discrete-event simulation on the Internet and the web. In P. Fishwick, D. Hill, and R. Smith, editors, *The International Conference on Web-Based Modelling and Simulation*, 30, pages 57–62, San Diego, California, January 1998. The Society for Computer Simulation International.

[157] E. H. Shortliffe. *Medical Informatics Meets Medical Education.* http://www-smi.stanford.edu/smi/academics/jama-pulse.html.

[158] J.C. Silverstein. *Web-based Segmentation and Display of Three-dimensional Radiologic Image Data.* http://www.sbhis.uic.edu/jsilver/ Web-Seg/index.htm.

[159] A. Smith, Y. Kitamura, and et.al. A simple and efficient method for accurate collision detection among deformable polyhedral objects in arbitrary motion. In *Proceeding of IEEE Annual International Symposium of Virtual Reality*, pages 136–145, Triangle Park, March 1995.

[160] T. Smith, J. Shi, and et.al. JackMOO: A web-based system for virtual human simulation. In P. Fishwick, D. Hill, and R. Smith, editors, *The International Conference on Web-Based Modelling and Simulation*, 30, pages 145–52, San Diego, California, January 1998. The Society for Computer Simulation International.

[161] Z. Soferman, D. Blythe, and N. John. Advanved graphics behind medical virtual reality: Evolution of algorithms, hardware and software interfaces. *Proceedings of the IEEE*, 86(3):531–554, 1998.

[162] I.S. Sokolnikoff. *Mathematical Theory of Elasticity.* McGraw-Hill Book Company Inc., New York, second edition, 1956.

[163] G. Song and N. Reddy. Towards virtual reality of cutting: A feasibility study. In *Proceeding of the 16th Annual International Conference*, 2, pages 1037–8, Baltimore, November 1994.

[164] G. Song and N. Reddy. Tissue cutting in virtual environments. In K. Morgan and R. Satava, editors, *Interactive Technology and New Paradigm for Healthacre*, pages 359–64. IOS Press and Ohmsha, 1995.

[165] K. Stanney. Realising the full potential of virtual reality: Human factors issues that could stand in the way. In *Proceeding of IEEE Annual International Symposium of Virtual Reality*, pages 28–34, Triangle Park, March 1995.

[166] D. Stredney. Virtual simulations through high performance computing. In *Medicine Meets Virtual Reality II*, pages 212–15, San Diego, January 1994.

[167] N. Taffinder, C. Sutton, R. Fishwick, and et.al. Validation of virtual reality to teach and assess psychomotor skills in laparoscopic surgery: Results from randomised controlled studies using the MIST VR laparoscopic simulator. In J.D. Weswood, H.M. Hoffman, and et.al., editors, *Medicine Meets Virtual Reality*, pages 124–9. IOS Press and Ohmsha, 1998.

[168] R. Tam, C. Healey, B. Flak, and et. al. Volume rendering of abdominal aortic aneurysms. In *Proceedings IEEE Visualization '97*, pages 136–145, 1997.

[169] A. Tanenbaum. *Computer networks.* Prentice Hall, Upper Saddle River, N.J., 1996.

[170] D. Terzopoulos. Deformable models: Formulations and applications. In D. Thalmann N. Magnenat-Thalmann, editor, *New Trends in Animation and Visualization*, chapter 19, pages 273–284. Wiley, 1991.

[171] D. Terzopoulos. On deformable models. In P. Concus, R. Finn, and O. Hoffman, editors, *Geometric Analysis and Computer Graphics*, pages 181–92. Spring Verla, 1991.

[172] D. Terzopoulos and K. Fleischer. Modelling inelastic deformation: Viscoelasticity, plasticity, fracture. *Computer graphics*, 22(4):269–78, 1988.

[173] D. Terzopoulos and D. Metaxas. Dynamic 3D models with local and global deformation: Deformable superquadrics. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(7):703–14, 1991.

[174] D. Terzopoulos, J. Platt, and et.al. Elastically deformable models. *Computer Graphics*, 21(4):205–214, 1987.

[175] D. Terzopoulos and A. Witkin. Physically based models with rigid and deformable components. *IEEE Computer Graphics and Applications*, 8(6):41–51, 1988.

[176] N. M. Thalmann and Y. Yang. Techniques for cloth animation. In D. Thalmann N. Magnenat-Thalmann, editor, *New Trends in Animation and Visualization*, chapter 17, pages 243–55. Wiley, 1991.

[177] X. Tu and D. Terzopoulos. Artificial fishes: Physics, locomotion, perception, behavior. In *Computer Graphics Proceedings*, pages 43–49, Florida, July 1994.

[178] C. Varela, D. Nekhayeu, and et.al. DB: Browsing object oriented databases over the web. In *Fourth International World Wide Web Conference : the Web revolution*, pages 209–20, Boston, Massachusetts, December 1995. International World Wide Web Conference, O'Reilly and Assoc.

[179] P. Volino, M. Courchesne, and N. Thalmann. Versatile and efficient techniques for simulating cloth and other deformation objects. In *Computer Graphics Proceedings*, pages 137–44, Los Angeles, California, August 1995.

[180] T. Walsam, K. Zuiderveld, and et. al. CT-based simulation of fluoroscopy and DSA for endovascular surgery training. In *Proceedings CVRMed- MRCAS'97*, volume 1205, pages 273–282, 1997.

[181] F.Y. Wan. *Introduction to the Calculus of Variation and its applications.* ITP, New York, 1995.

[182] Kang Wang, Isaac Kohane, and et. al. *A Real Time Patient Monitoring System on the World Wide Web.* http://icu.lcs.mit.edu/imidemo/, 1996.

[183] A. Witkin. An introduction to physically based modelling: Particle collision and contact. In *SIGGRAPH'97*, 1997.

[184] A. Witkin and D. Baraff. An introduction to physically based modelling. In *SIGGRAPH'97*, 1997.

[185] A. Witkin, K. Fleischer, and A. Barr. Energy constraints on parameterised models. *Computer Graphics*, 21(4):225–232, 1987.

[186] A. Witkin and W. Welch. Fast animation and control of non rigid structures. *Computer Graphics*, 24(4):243–52, 1990.

[187] M. Wloka. Interacting with virtual reality. In J. Rix, S. Haas, and J. Teixeira, editors, *Virtual Prototyping: Virtual Environments and the Product Development Process.* Chapman & Hall, 1995.

[188] Jason Wood, Ken Brodlie, and Helen Wright. Visualization over the World Wide Web and its application to environmental data. In *IEEE Visualization '96*, San Francisco, October 1996. http://cself27.leeds.ac.uk:8010/jason/html/pollution.html.

[189] K. Wright. *Practical Training in Interventional Radiology.* http://www.medic-online.net/ir/index.html.

[190] Y. Wu, D. Thalmann, and N. Thalmann. Deformable surfaces using physically based particle systems. *Computer Graphics: Developments in Virtual Environments*, pages 205–15, 1995.

[191] R. Yagel, D. Stredney, and et.al. Building a virtual environment for endoscopic sinus surgery simulation. *Computers and Graphics*, 20(6):813–23, 1996.

[192] T. Yamaguchi, N. Furuta, and et.al. The hyper hospital - a novel medical care system on the virtual reality network. In *Proceedings of the 16th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, 16, pages 1388–9, 1994.

[193] R. Ziegler. Haptic displays - how can we feel virtual environments. *State-of-the-Art report*, 1996. http://www.igd.fhg.de/www/igd-a4/reports/index_96_e.html.

[194] R. Ziegler, W. Mueller, and et.al. A virtual reality medical training system. In Nicholas Ayache, editor, *International Conference on Computer Vision, Virtual Reality, and Robotics in Medicine, CVR Med '95*, pages 282–86, Nice, France, 1995. Springer-Verlag.