Simulation of image guided needle puncture: contribution to real-time ultrasound and fluoroscopic rendering, and volume haptic rendering

Franck P. Vidal

PRIFYSGOL
BANGOR
UNIVERSITY
1884

Thesis Submitted in Candidature for the Degree of Doctor of Philosophy

January 2008

School of Computer Science
Bangor University
United Kingdom

ABSTRACT

The potential for the use of computer graphics in medicine has been well established. An important emerging area is the provision of training tools for interventional radiology (IR) procedures. These are minimally invasive, targeted treatments performed using imaging for guidance. Training of the skills required in IR is an apprenticeship which still consists of close supervision using the model i) see one, ii) do one and iii) teach one. Simulations of guidewire and catheter insertion for IR are already commercially available. However, training of needle guidance using ultrasound (US), fluoroscopic or computerized tomography (CT) images, the first step in approximately half of all IR procedures, has been largely overlooked.

This thesis is devoted to the development of novel techniques for the simulation of IR procedures. The result is the realization of an integrated visual-haptic system, called *BIGNePSi*, for the simulation of US guided needle puncture using patient specific data with 3D textures and volume haptics. The system is a cost effective training tool, using off-the-shelf components (visual displays, haptic devices and working stations), that delivers a high fidelity training experience.

We demonstrate that the proxy-based haptic rendering method can be extended to use volumetric data so that the trainee can feel underlying structures, such as ribs and bones, whilst scanning the surface of the body with a virtual US transducer. A volume haptic model is also proposed that implements an effective model of needle puncture that can be modulated by using actual force measurements. A method of approximating US-like images from CT data sets is also described. We also demonstrate how to exploit today's graphics cards to achieve physically-based simulation of x-ray images using GPU programming and 3D texture hardware. We also demonstrate how to use GPU programming to modify, at interactive framerates, the content of 3D textures to include the needle shaft and also to artificially add a tissue lesion into the dataset of a specific patient. This enables the clinician to provide students with a wide variety of training scenarios.

Validation of the simulator is critical to its eventual uptake in a training curriculum and a project such as this cannot be undertaken without close co-operation with the domain experts. Hence this project has been undertaken within a multi-disciplinary collaboration involving practising interventional radiologists and computer scientists of the Collaborators in Radiological Interventional Virtual Environments (CR*a*IVE) consortium. The cognitive task analysis (CTA) for freehand US guided biopsy performed by our psychologist partners has been extensively used to guide the design of the simulator. To further ensure that the fidelity of the simulator is at an acceptable level, clinical validation of the system's content has been carried out at each development stage.

In further, objective evaluations, questionnaires were developed to evaluate the features and the performances of the simulator. These were distributed to trainees and experts at different workshops. Many suggestions for improvements were collected and subsequently integrated into the simulator.

RÉSUMÉ

Le potentiel de l'informatique graphique (ou infographie) en médecine est désormais bien établi. Un important domaine qui émerge actuellement est le développement de logiciels d'entraînement pour les procédures de radiologie interventionnelle (RI). Ces dernières sont des traitements, minimalement invasifs, guidés par des images médicales. L'entraînement des compétences requises en RI est un apprentissage qui consiste toujours en la supervision par un expert selon le modèle : i) observer, ii) reproduire et iii) montrer. Les simulations de l'insertion de fils guides et de cathéters pour la RI sont déjà disponibles commercialement. Cependant, l'entraînement de la ponction guidée par des images ultrasoniques, fluoroscopiques ou tomographiques, la première étape d'environ la moitié de toutes les procédures en RI, a largement été oublié.

Cette thèse est dévouée au développement de techniques novatrices pour la simulation de procédures de RI. Le résultat est la réalisation d'un système visuo-haptique intégré, appelé *BIGNePSi*, pour la simulation de ponction guidée par ultrasons en utilisant des données spécifiques d'un patient avec des textures 3D et un rendu haptique volumique. Le système est abordable financièrement, en utilisant des composants disponibles commercialement et relativement bons marchés (dispositif d'affichage, périphériques haptiques et stations de travail), et offre une expérience d'entraînement hautement fidèle à la réalité.

Nous démontrons que le rendu haptique basé sur la notion de “proxy” peut être étendu en utilisant des données volumiques de façon à permettre à l'utilisateur de percevoir les structures internes, telles que les vertèbres et les os, pendant qu'il scanne la surface du corps du patient avec la sonde à ultrasons virtuelle. Un algorithme de rendu haptique volumique implémentant un modèle efficace de ponction et pouvant être modulé grâce des mesures expérimentales est aussi proposé. Une méthode pour simuler des images ultrasoniques à partir de données tomographiques est aussi décrite. Nous démontrons aussi comment exploiter les capacités des cartes graphiques actuelles pour simuler, selon les lois de la physique, des images fluoroscopiques en utilisant des textures 3D ainsi que la programmabilité des processeurs graphiques (GPUs). Nous démontrons aussi comment utiliser la programmabilité des GPUs pour modifier en temps réel le contenu des textures 3D pour inclure l'aiguille et aussi pour ajouter artificiellement une lésion des tissues dans les données spécifiques d'un patient. Ceci permet au clinicien de fournir aux étudiants une grande variété de scénarii d'entraînement.

La validation du simulateur est un point critique pour son éventuelle utilisation en formations hospitalières. Un projet tel que celui-ci ne peut donc être réalisé sans une proche collaboration avec les experts médicaux de ce domaine. Ce projet est donc le fruit d'une collaboration multidisciplinaire, incluant des radiologistes interventionels et des informaticiens du consortium Collaborators in Radiological Interventional Virtual Environments (CR*a*IVE). L’analyse cognitive de la ponction guidée par ultrasons a été réalisée par nos partenaires psychologistes. Cette analyse a été utilisée intensivement comme aide au design du simulateur. En addition, pour s’assurer que le simulateur reproduit la tache à accomplir à un niveau acceptable, la validation clinique du contenu du système a été réalisée à chaque étape du développement. En addition, pour l’évaluation objective des fonctionnalités et des performances du simulateur, des questionnaires ont été développés. Ces questionnaires ont été distribués à des étudiants et des experts en radiologie interventionnelle à différents séminaires. De nombreuses suggestions pour l’amélioration du système ont ainsi été collectées et intégrées progressivement dans le simulateur.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LISTINGS

ACKNOWLEDGMENTS

I would like to express my sincere gratitude to all the people who made this thesis possible.

Special thanks go to Professor Nigel W. John, for his support and trust over the time of research and for his corrections of this thesis.

Many thanks to Dr Derek A. Gould, consultant interventional radiologist at Royal Liverpool University NHS Trust, whose medical expertise, stimulating suggestions and encouragement helped me during the time of research.

Thanks to Dr Dare Seriki, consultant interventional radiologist at Royal Preston Hospital, for inviting me to present my research during three annual training days.

I also would like to thank interventional radiologist fellows at Liverpool, Andrew E. Healey for supplying me with experimental force measurements, Steven Powell and Darshan Bakshi for their help and enthusiasm during this work.

I also acknowledge Dr. Nicholas Chalmers, interventional radiologist at Manchester Royal Infirmary, for dealing with ultrasound images and CT data sets, and for his advices at the early stage of this research.

I also would like to thank the people of the School of Informatics for welcoming me into their department, particularly German, Jon, Min, Oscar, and Yanhua, who are always up for a cup of tea or a pub meal. Thanks also to the technical and administrative staff of the department for their support.

Many thanks go to all the friends I met during these years, who made my stay at Bangor enjoyable. To Anoop, Chris, and Llyr for sharing the highs and lows of this experience; to Ade for so many UT2004 CTF sessions; Ik Soo for his discussions; and the other people of the High Performance Visualization and Medical Graphics group for welcoming me into their group. To the French *stagiaires* - Thomas Q., Alexandre S., Alexandre P., Fabien, Thomas E., Johann, Guillaume, Stéphane, Jérémie and Manuel - for laughs, pétanque games and imported raclette. To Fatima for those coffees and our expeditions around Bangor. To Paul, Michèle, and Zhaoxia, for climbing evenings and hill walking; to Jen, Lindsay, Terry, Ken, Izzy, Andy L., Andy H., M@, Laurie, Andy S., Jonathan, Mandy, Andrea, Knut, Janet, ... for all those moments walking up mountains in any weather conditions (of course preferably Welsh rain); and to all those who are not explicitly named but should be.

Je remercie mes parents et ma famille, pour leur soutient moral infaillible. Sans vous, cette thèse aurait été mission impossible, merci.

Statement of Originality

The work presented in this thesis was carried out by the candidate, except where otherwise stated. It has not been presented previously for any degree, nor is it at present under consideration by any other degree awarding body.

Candidate:

Franck P. Vidal

Statement of Availability

I hereby give consent for my thesis, if accepted, to be available for photocopying and for inter-library loan, and the title and summary to be made available to outside organisations.

Candidate:

Franck P. Vidal

Chapter 1

INTRODUCTION

Contents

1.1 Context

Over the past three decades computer graphics and visualization have played a growing role in adding value to a wide variety of medical applications. The earliest examples were reported in the mid 1970's when three-dimensional visualizations of computerized tomography (CT) data were first reported [209, 96]. Today, a variety of imaging modalities are in common use by the medical profession for diagnostic purposes, and these modalities provide a rich source of data for further processing using computer graphics and visualization techniques, a comprehensive review of this subject is given in [199]. Applications include medical diagnosis, procedures training, pre-operative planning, telemedicine and many more [161, 264]. A survey of the current state-of-the-art focused on interventional radiology (IR) can be found in Chapter 3. Indeed, the provision of training tools for IR procedures is an important emerging area [44, 83]. Interventional radiology is a core skill for many minimally invasive procedures that make use of imaging to guide catheters (tubes) and wires through organ systems using a small portal of entry (minimal access) into the body. As a result, IR techniques generally have less risk, less postoperative pain and shorter recovery time compared to open surgery. Training of needle puncture guided by imaging (*e.g.* ultrasound [122], computerized tomography [121] or any other imaging modality) using the Seldinger technique [232] - the first step in approximately half of all IR procedures - has been largely overlooked.

1.2 Hypothesis

Effective training of ultrasound guided needle puncture leads to many challenges. We summarise these questions, facts and aims with our thesis hypothesis.

> *A flexible and cost effective integrated visual-haptic system for the simulation of Ultrasound Guided Needle Puncture, can be realized using off-the-shelf components (Visual Displays, Haptic Device and Working Stations), to deliver a high fidelity training experience.*

The Bangor Image Guided NEedle Puncture SImulator (BIGN*e*PS*i*) project is an investigation of this hypothesis.

1.3 Scope and Contribution

Given the previous sections, this thesis relates the problems involved in the development of a cost effective virtual environment for training ultrasound guided needle puncture. It will focus on the major contribution of this work:

Needle Tracking And Adding Tissue Lesions in 3D Textures We propose a new technique to update, using GLSL, the 3D texture corresponding to a patient CT volume dataset to track the needle in realtime (see Section 4.4 for details). For the rendering of ultrasound or x-ray images, full advantage is taken of 3D texture hardware. It is not possible to update the content of the 3D texture in realtime to change the voxel intensities at the needle position because of the overhead involved in transferring the recalculated data from main memory to the graphics card. An important new development in graphics programming, however, is the use of high-level programming languages to take control over critical stages of the graphics rendering pipeline. Using the new OpenGL Shading Language, it is now possible to avoid the limitations of the fixed graphics rendering pipeline and to take advantage of programmable graphics hardware. Using GLSL, it becomes possible to detect if the texture coordinates corresponding to each pixel of the image to render, computed from the 3D texture, are being penetrated by the needle. Tissue lesions can also be added at any position within the patient dataset using a similar technique (see Section 4.5 for details).

Approximating Ultrasound Images We introduce a new graphics rendering pipeline to approximate in realtime convincing ultrasound images from scanned CT data (see Section 4.6 for details). Simulating ultrasound images in realtime is a difficult task because of the conflicting demands of interactivity and accuracy. Simulators reproducing the physics of sound waves have been available for decades [108]. Although computer hardware gets faster, physically-based simulation still remains a demanding task in term of computational time. Thus such an approach is not adequate for the use in a real-time virtual environment for training. A possible alternative approach is to transform CT data by adding artefacts characteristic of US images, such as acoustic shadowing and sound reflection at interfaces [267].

X-ray Images Simulation We also propose a volume rendering method to achieve, on the GPU and in realtime, physically-based simulation of fluoroscopic (*i.e.* x-ray) images from CT data sets (see Section 4.7). The main approach to guide needles is by using US images. Nevertheless, fluoroscopy can also be used to guide nephrostomy needles and catheters, but then a contrast medium is introduced into the collecting system of the kidney, outlining on the fluoroscopic images the fluid filled elements that the needle must be guided into [248]. Simulating x-ray images using the physical laws has been being extensively studied in physics. Real-time performance using geometrical or CAD models can be almost achieved when only the attenuation of x-rays is computed, *i.e.* without taking into account photons/matter interactions such as absorbed photons or scattered photons. The physically-based simulation of x-ray images in realtime from volume data sets still remains a challenge. In this thesis, we demonstrate how to exploit the capability of today's graphics cards to efficiently achieve this [266].

US Probe Haptic Model We describe a new hybrid surface/volume haptic rendering to provide the realistic manipulation of the virtual ultrasound probe by combining the classic proxy-based algorithm [215] with patient specific CT datasets to feel internal hard structures, such as ribs, when scanning the patient's body (see Section 5.3.1).

The use of haptics is an important component in providing realistic manipulation of the virtual tools being used, here a needle and a image probe. The classic proxy-based algorithm is commonly accepted for the haptic rendering of surfaces. In this model, the haptic device interacts with the surface of a 3D object represented as a triangular polygon mesh. A small sphere, the virtual proxy, represents the desired position of the device. The proxy always remains outside of the surface whereas the device can penetrate into the object with which it is interacting. The surface will constrain the haptic device to reach the proxy. This model does not natively take into account internal hard structures, such as bony landmarks in the case of the human body, when scanning the surface. Using the voxel values in the CT volume dataset, it is possible to adapt the proxy-based algorithm to modify the stiffness of the scanned object according to the materials directly located below the skin at the contact with the ultrasound probe. For example, the stiffness will be higher if a rib is just below the ultrasound probe, lower in the case of fat.

Needle Haptic Rendering We describe a new volume haptic model based on experimental data to provide the realistic manipulation of the virtual needle (see Section 5.3.2).

Look up tables (LUTs) containing a sequence of empirically defined values is one approach that has been applied to needle insertion [79]. Each entry in the LUT contains the force value that is applied once the needle has reached a particular depth. Physically-based models that take into account the elastostatic properties of tissues under distortion have also been reported [48, 49]. Recent work has focused on the measurement of forces during procedural instrumentation of real patients [93]. The data can be used to create and validate a haptic model. Our system uses two different haptic models to mimic the needle manipulation. Before penetrating the skin surface, the classic proxy-based algorithm [215] can be used, but with different friction parameters to mimic a needle moving over the skin surface. When the force applied by the needle at the insertion point exceeds a given threshold, the haptic rendering mode is switched to a secondary mode based on Hooke's law that allows the needle to penetrate internal tissues [267].

1.4 List of Publications

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

1. **F. P. Vidal**, N. W. John, A. E. Healey, and D. A. Gould. Simulation of Ultrasound Guided Needle Puncture using Patient Specific Data with 3D Textures and Volume Haptics. *Computer Animation and Virtual Worlds*, Wiley & Sons, Ltd., to appear, DOI: 10.1002/cav.217

2. **F. P. Vidal**, N. W. John, and R. M. Guillemot. Interactive Physically-based X-ray Simulation: CPU or GPU? In *Proceeding of Medicine Meets Virtual Reality 15 - in vivo, in vitro, in silico: Designing the Next in Medicine (MMVR15)*, poster, Long Beach, California, 6 - 9 February 2007. Volume 125 of *Studies in Health Technology and Informatics*, IOS Press, 2007, pages 479–481, PMID: 17377331

3. **F. P. Vidal**, F. Bello, K. W. Brodlie, D. A. Gould, N. W. John, R. Phillips, and N. J. Avis. Principles and Applications of Computer Graphics in Medicine. *Computer Graphics Forum*, Blackwell, 25(1):113–137, 2006, DOI: 10.1111/j.1467-8659.2006.00822.x

4. **F. P. Vidal**, N. Chalmers, D. A. Gould, A. E. Healey, and N. W. John. Developing a Needle Guidance Virtual Environment with Patient Specific Data and Force Feedback. In *Proceeding of the 19th International Congress and Exhibition of Computer Assisted Radiology and Surgery (CARS '05)*, Berlin, Germany, 22 – 25 June 2005. Volume 1281 of *International Congress Series*, Elsevier, 2005, pages 418–423, DOI: 10.1016/j.ics.2005.03.200

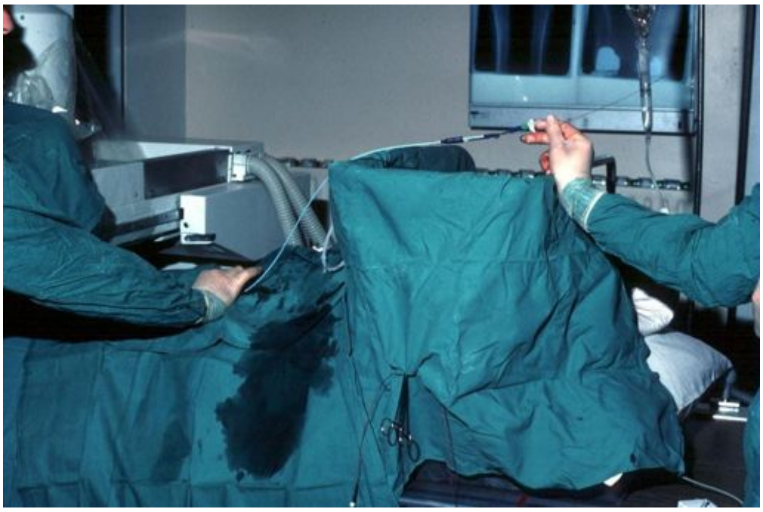
5. **F. P. Vidal**, F. Bello, K. Brodlie, N. W. John, D. Gould, R. Phillips, and N. Avis. Principles and Applications of Medical Virtual Environments. In *STAR Proceedings of Eurographics 2004 (EG '04)*, Grenoble, France, 30 August – 3 September 2004, Eurographics Association, 2004, pages 1–35.

1.5 Outline

This thesis describes research carried out in the field of medical virtual environment, particularly for ultrasound guided needle puncture in interventional radiology. Chapter 2 gives the medical context and provides an overview of interventional radiology procedures. Chapter 3 provides a brief overview of the most common medical image modalities. The mathematical and physical theory underlying ultrasound, x-ray and computerized tomography imaging, the modalities of interest in this doctoral study, are also presented. A review of previous work covers developments of the use of medical visualization and virtual environments, focusing on the techniques applied and their applications. A detail description of the algorithms implemented using programmable graphics processing unit (GPU) are given in Chapter 4. It includes i) approximating US images (see Section 4.6), ii) physically-based simulation of x-ray images from CT datasets using 3D texture hardware (see Section 4.7), iii) updating 3D textures at interactive frame rate to track the needle's position and orientation (see Section 4.4), and iv) updating 3D textures to add tissue lesions into the patient's data (see Section 4.5). The volume haptic models developed to mimic the virtual needle and the virtual image probe are presented in Chapter 5. It includes i) a hybrid surface/volume haptic rendering model to mimic an ultrasound transducer (see Section 5.3.1), and ii) a volume rendering model to mimic needle puncture (see Section 5.3.2). Chapter 6 describes the realisation and the results within the simulator of the algorithms presented in the previous chapters. The last chapter of this thesis discusses the work carried out and provides some possible directions for further work.

Chapter 2

MEDICAL PROCEDURE



Contents

2.1 Interventional Radiology

interventional radiology (IR) is a core skill for many minimally invasive procedures that make use of imaging to guide catheters (tubes) and wires through organ systems using a small portal of entry (minimal access) into the body. As a result, IR techniques generally have less risk, less postoperative pain and shorter recovery time compared to open surgery. The picture on page 7 shows a catheter being passed over wire into a vessel. In IR, the practitioner does not look at his/her hands but at a monitor where images are displayed (see Figure 2.1).

IR is a quite recent speciality founded on a pioneering work by Charles Dotter, a radiologist at the University of Oregon in Portland, USA, who performed the first “transluminal angioplasty” during a diagnostic arteriogram in 1964 [52]. Entry to the body is usually by means of the Seldinger technique, developed by Sven-Ivar Seldinger in 1953 [232]. This technique was initially used only for diagnostic imaging. Angioplasty was then further developed to open up blocked arteries, to treat patients with peripheral vascular disease. The introduction of the balloon angioplasty catheter in 1972 by Andreas Gruntzig paved the way for a far less invasive approach than traditional vascular surgery. Since these early days, IR extends to a vast range of imaging guided procedures which are minimally invasive. Targeted treatments are performed using imaging for guidance in all organ systems, including the liver, the urinary tract, biliary system, gut and blood vessels in the field of many different medical and surgical specialities [167]. These procedures include angioplasty, embolisation, thrombectomy, biopsy, nephrostomy, percutaneous biliary access, aneurysm endograft, vertebroplasty, gastrostomy, radiofrequency tumour ablation. Fluoroscopic imaging, CT, ultrasound (US), or other imaging modalities are used for guidance. US is particularly commonly used for biopsy [245, 160] and nephrostomy [189]. More recently, real-time magnetic resonance imaging (MRI) has also been applied for IR guidance [33].

2.2 Freehand Ultrasound Guided Needle Puncture

According to Otto, ultrasound-guided fine-needle abdominal biopsy was first described in 1972 [186]. Matalon and Silver reviewed the use of ultrasound images as a guidance of interventional procedures in the early 90s, and discussed the advantages of US guidance over CT guidance [160]. Finally, they explain how to place a needle into a target under US guidance using the freehand technique. More recently, Chapman *et al* presented the basics of ultrasound guided puncture, emphasising how to visualize the needle, and particularly directed at anaesthetists and intensivists rather than interventional radiologists [35].

During the project, to identify, describe and detail the whole process of core interventional radiology procedures, from the cues perceived, to the cognitive thought processes of decision-making, to the ensuing physical actions, cognitive task analysis (CTA) studies have been performed by our psychologist partners within the Collaborators in Radiological Interventional Virtual Environments (CR*a*IVE) consortium [41, 123]. This work includes CTA for freehand ultrasound guided biopsy [122], and it has been used extensively to guide the design of the simulator described in this thesis. An excerpt from this detailed task analysis is included in Appendix A.

Generally, an important concept of ultrasound-guided procedures is real-time interaction. To ensure the correct placement of the needle into the target, the practitioner must establish an interactive feedback loop between the real-time US images and the needle placement (see Figure 2.1). Figure 2.2 illustrates a possible placement of the needle for the freehand technique. However, US guided techniques have been reported as being underused because of the lack of understanding of real-time ultrasound images, in particular to locate the position of the needle tip [160, 35].

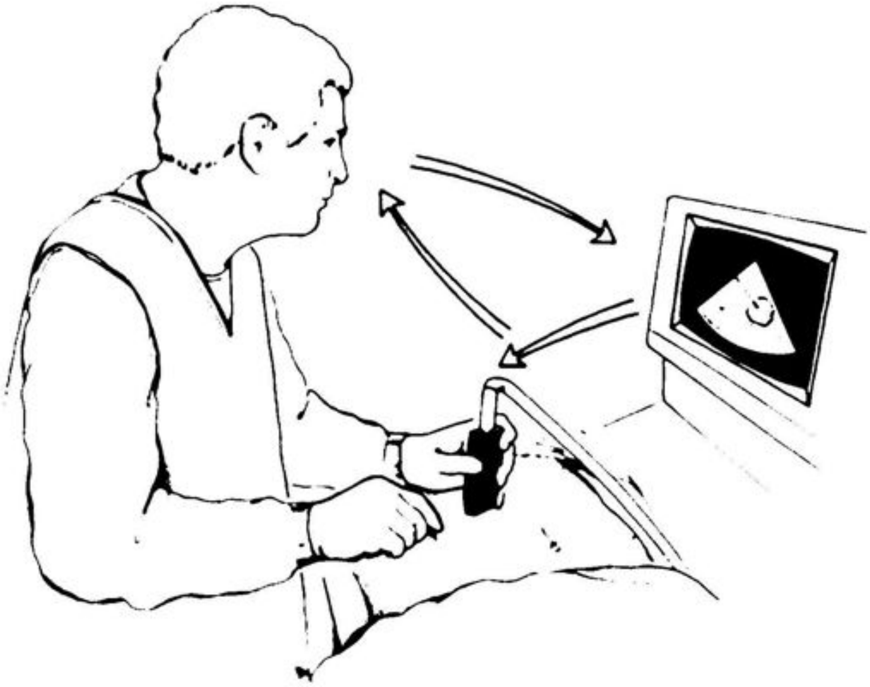


FIG. 2.1: *Interactive feedback loop between the real-time ultrasound images and the needle placement.* Source*: Reproduced from [160], published by the Radiological Society of North America.*

Figures 2.4(b), 2.6(b), 4.10, 6.8(c), and 6.8(d) are examples of actual ultrasound images.

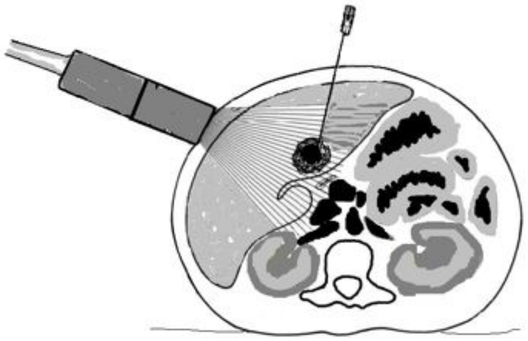


FIG. 2.2: *Placement of the needle using ultrasound guidance and the freehand technique. Image courtesy of Dr Derek A. Gould at Radiology Department, Royal Liverpool Trust.*

2.3 Motivations for a Simulator of Ultrasound Guided Needle Puncture

2.3.1 Current Training Methods

Most of the training of the visual and motor skills required in IR is performed as a traditional apprenticeship, which today still consists of close expert supervision in patients using the model i) see one, ii) do one and iii) teach one, *i.e.* the trainee first observes a particular procedure, then he/she participates more and more under the supervision of the expert, to finally become an expert who will teach the skill to a new trainee, *etc.* (see Figure 2.3).

FIG. 2.3: *The Anatomy Lecture of Dr. Nicolaes Tulp, by Rembrandt, 1632.*

2.3.2 Drawback of the Current Training Methods

In apprenticeship training methods, procedures are learnt and skills obtained during practice in patients. Training in this way may, however, subject patients to increased discomfort, greater risk of complication, and prolonged procedure times, creating a clinical governance dilemma within the modern healthcare environment. At the same time, there is limited access to apprenticeship training in more complex scenarios [212] with inevitable difficulty training in a time efficient manner [27].

There is currently a shortage of radiology consultants, and specifically those specialised in IR, both within and outside the UK. Limitations in the current processes of apprenticeship training [55], together with the introduction of the Calman system in the United Kingdom [29, 244, 173] and the implementation of regulatory restrictions on working hours during training years [112, 47], are driving a need for a fresh approach to teaching clinical skills.

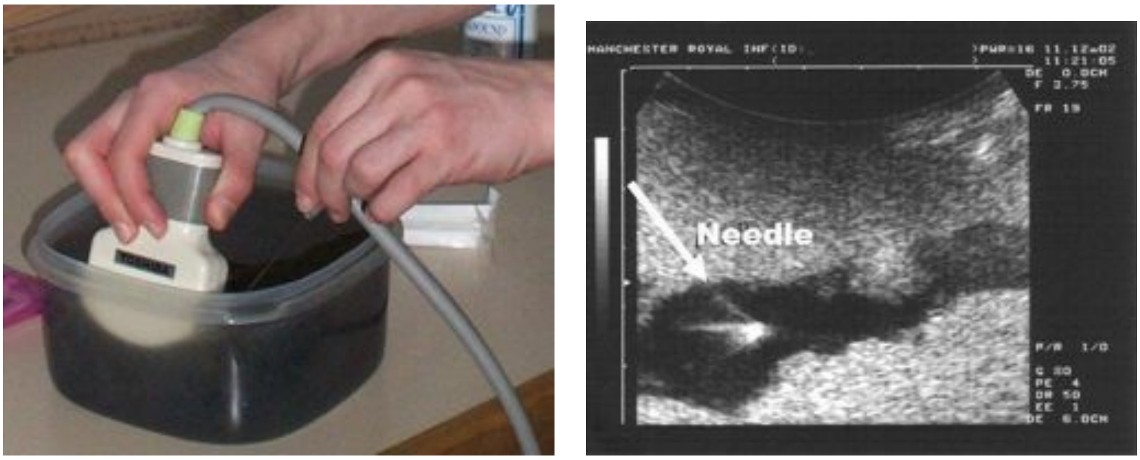
The Cardiovascular and Interventional Radiological Society of Europe (CIRSE), the Society of Interventional Radiology (SIR) and the Radiological Society of North America (RSNA) recognise the current shortfall in evidence and have recently established individual medical simulation task forces, and a joint task force. They have set out joint recommendations, also supported by the British Society of Interventional Radiologists (BSIR), on the development and use of medical simulation to train and assess IR [83]. They state that contemporary simulators are suitable for gaining certain aspects of procedural experience, such as learning the correct sequence of procedural steps and selection of appropriate tools: many medical errors result from incorrect procedural sequencing. Whilst acknowledging that this may be beneficial prior to performing procedures on patients, they caution that the utility of simulators for the acquisition of catheter manipulation skills is currently unproven and that experience on a simulator cannot yet be regarded as equivalent to training involving performance of actual endovascular procedures in patients. Test validation is recommended to include content, construct, concurrent, and predictive validation with the objective of demonstrating transfer of trained skills to procedures in patients.

2.3.3 Current Alternative Training Methods

Current possible training alternatives include various simulations which use mannequins [153], animals [157, 51] and computer-based methods [264]. An important emerging area is the provision of training tools for interventional radiology procedures. In using some form of simulation to replace patients, computer based simulations of guidewire and catheter insertion for IR are already available commercially, and the best examples are discussed in Section 3.4.

However, training of needle puncture guided by imaging (*e.g.* ultrasound [122], computerized tomography [121] or any other imaging modality) using the Seldinger technique - the first step in approximately half of all IR procedures - has been largely overlooked. Some hospitals have adopted ‘low tech’ but reasonable effective home grown solutions [70, 163]:

1. The Manchester Royal Infirmary, for example, makes use of a phantom model built from a plastic box filled with foam into which tube like structures have been hollowed out in order to represent blood vessels. Water is added to the box, which is then placed into a microwave oven for several minutes to remove all of the air bubbles. The phantom model can then be imaged using a conventional ultrasound scanner, and produces images similar in appearance to the ultrasound images of real patients that the clinician will see in practice (see Figure 2.4).



(a) The trainee is holding the US probe with his right hand and inserting the needle using his left hand.

(b) Example of ultrasound image produced.

FIG. 2.4: *Home-made simple environment for training needle guidance with ultrasound at the Manchester Royal Infirmary. Images courtesy of Dr Nicholas Chalmers at Manchester Royal Infirmary.*

2. Figure 2.5 is another example of home-made mannequin. It was used during an annual Interventional study day, at the Royal Preston Hospital, to teach IR students how to perform ultrasound guided puncture. It consists of a plastic box filled with agar. Water is added to

the box. Finally, olives and grapes are hidden in the box below the agar. They correspond to targets that trainees will have to puncture.

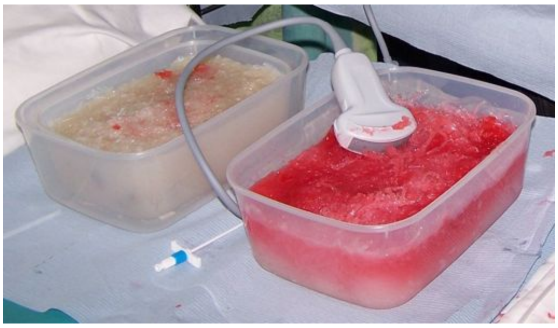


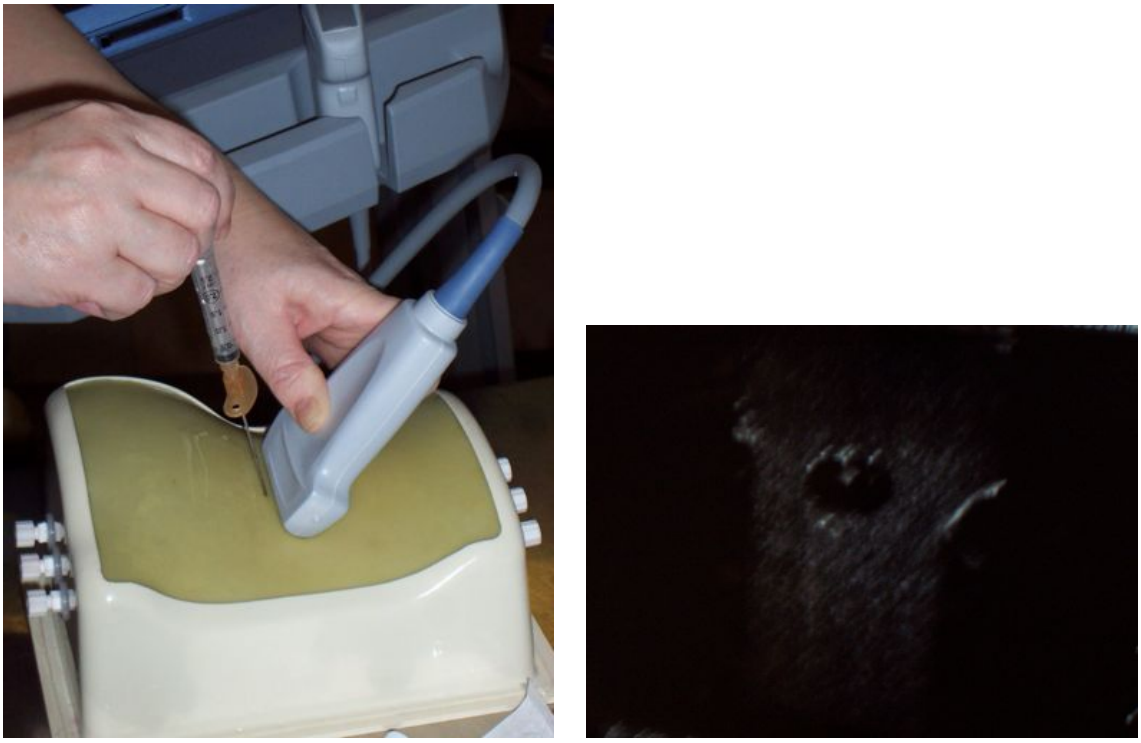
FIG. 2.5: *Home-made mannequin for ultrasound guided needle puncture in use during an Interventional study day at the Royal Preston Hospital.*

3. Figure 2.6 is an example of the phantom used at the Royal Liverpool University National Health Service Trust for training needle insertion guided by ultrasound. The model is a cloth sewn positive of a vessel trifurcation filled with low melting point alloy. The cloth is cut away. The alloy is then sprayed with texoflex. Then, in a water bath, the alloy is melted away leaving only the texoflex vessel wall. This is then secured in a polyvinyl chloride box and filled with gelatin. Finally latex is place over the top to stop it from drying out.

These three home-made phantom models can be used with a conventional ultrasound scanner, and produces images similar in appearance to the ultrasound images of patients that the clinician will see in practice.

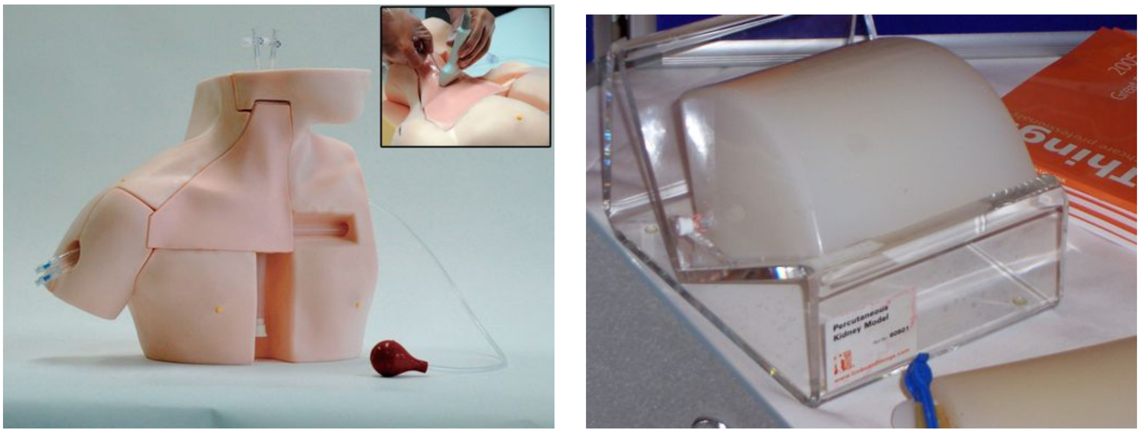
Figure 2.7 show proprietary models that use tissue mimics to provide effective simulation of ultrasound needle guidance, such as synthetic soft tissue models [138, 153]. However, they are quite expensive and wear out after multiple needle passes. Moreover, using solutions based on fixed anatomical models, it is difficult to provide sufficient variety of anatomical models to train the complete range of anatomical and pathological encounters, and the tactile properties and physiological responses of foam and other surrogate materials are far from that of a real person.

Anaesthetised animals such as pigs can also be used for training in IR [157, 51]. Although such trainings are performed under real conditions with the "patients" being alive, the use of animals for training IR procedures is restricted. Training this way is expensive (the animals may be killed after an operation), the human anatomy and pathology differ from any animals, and ethical and legal considerations must be taken into account when animals are used for training purpose. Indeed, it is not allowed to train in this way in some countries, such as the United Kingdom.



(a) (b)

FIG. 2.6: *a) Home-made mannequin for ultrasound guided needle puncture at the Royal Liverpool University NHS Trust; b) actual ultrasound image corresponding to Figure 2.6(a).*



(a) Phantom for training central venous catheter insertion. *Source:* Reproduced from [138], by Kyoto Kagaku Co., Ltd.

(b) Phantom for training percutaneous access of kidneys, by Limbs And Things.

FIG. 2.7: *Proprietary models for training ultrasound guidance of needles.*

Chapter 3

BACKGROUND AND RELATED WORK

Contents

3.1 Retrospective of Medical Imaging Techniques Used in IR

The interest for physicians to look inside the human body has existed since antiquity. Indeed the first reference of endoscopy corresponds to a description of the examination of the rectum with a speculum by Hippocrates (460-375 BC) [16]. Since then, much progress has been made. The examination of the internal organs and structures of the body using visible light has become commonplace and many other physical properties have now been adopted for imaging purposes.

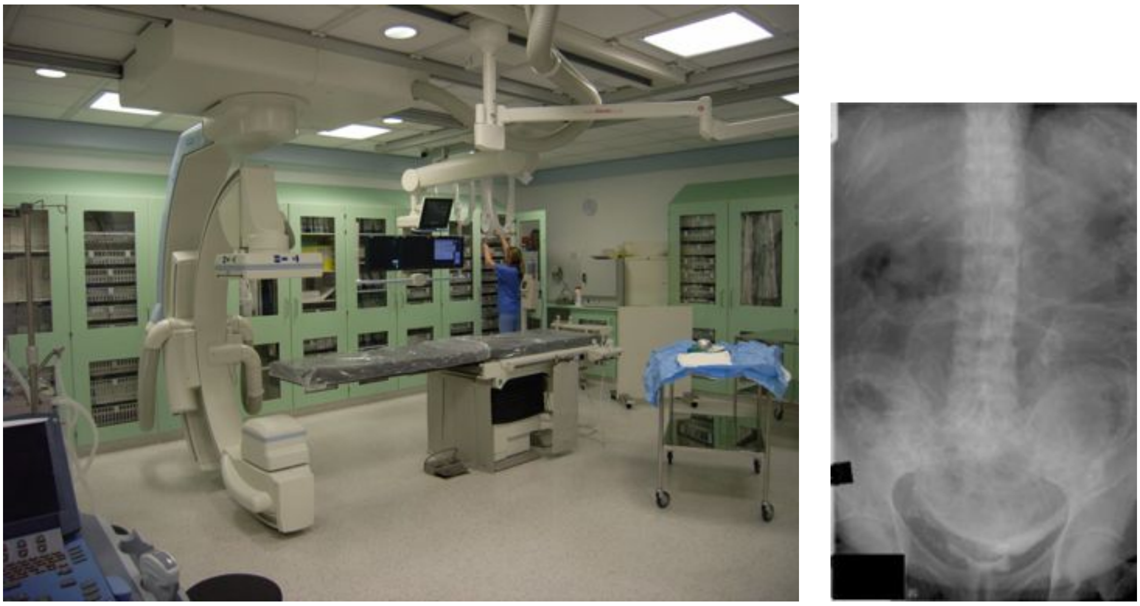
Section 3.1.1 highlights the evolution of x-ray imaging, from the first radiograph to micro-tomography. ultrasound is then reviewed in Section 3.1.2. Finally, the modality that has became more and more popular, magnetic resonance imaging (MRI) (formally known as nuclear magnetic resonance imaging (NMRI)), is presented in Section 3.1.3. Other image modalities exist such as fibre endoscopy [154, 16] and nuclear medicine imaging [9], but we focus here on those modalities most relevant to interventional radiology.

3.1.1 X-ray, Fluoroscopy, and X-ray Tomography

The first attempts to obtain images from inside the human body used visible photons, this technique is called endoscopy [154, 16]. X-ray imaging techniques make use of X photons, which are non-visible to the human eye. They were discovered in 1885 by Wilhelm Conrad Röntgen [116]. After demonstrating images of bones, he was awarded the Nobel prize in 1901 for the discovery of x-rays. Since then, the conventional x-ray radiograph has been extensively used as a diagnostic aid in medicine [285]. When x-rays are projected through the human body onto a photographic film or more recently, a digital detector, anatomical structures are depicted as a negative grey scale image according to the varying attenuation of the x-ray beam by different tissues. Conventional x-ray radiographs are still commonly used, for example in the diagnosis of trauma and chest disease.

X-rays are an electromagnetic radiation (like visible light), generally emitted when matter is bombarded with fast electrons (x-ray tubes). There are several techniques for generating x-rays (x-ray tube, linear accelerator, synchrotron, *etc.*), but x-ray tubes are still the main device used by radiographers. The first accelerators (cyclotrons) were built in the 1930's. Then, in 1947 in the USA, at General Electric, synchrotron radiation was observed for the first time [18, 279]. Initially, this effect was considered a nuisance because particles lose energy. Nevertheless, the flow of photons is much higher than in the case of x-ray tubes. Today, synchrotron radiation sources are compared to “super-microscopes” and are used in medical imaging [254], *e.g.* transvenous coronary angiography [56].

It is also possible to uses x-rays to produce real-time video images, this medical imaging modality is called fluoroscopy. Instead of using films, this is achieved using an image intensifier that converts x-rays into light. The light is then captured by a charge-coupled device (CCD) video camera allowing images to be displayed on a monitor. A C-arm fluoroscopic machine and a fluoroscopic image of an abdomen are shown in Figure 3.1. In this image, we can distinguish the femoral heads, the hip, the spinal column, ribs, and soft tissues all overlying each other. Landmarks in the evolution of Radiology include the first iodized contrast agent in 1943 which led to angiography, and the development of computerized tomography (CT) enabling acquisition of volumetric data of anatomical structures (see Section 3.2.1).



(a) C-arm fluoroscopic system.

(b) Fluoroscopic image of an abdomen.

FIG. 3.1: *Fluoroscopy. Images courtesy of Dr Derek A. Gould at Radiology Department, Royal Liverpool Trust.*

The first classical tomographic device was introduced in 1921 [19, 75]. Much later, Hounsfield successfully tested the first clinical Computerized Axial Tomography (CAT, generally reduced as Computerized Tomography, or CT) in 1972 [103]. Hounsfield received the Nobel Prize in Physiology or Medicine in 1979. Tomography is a non-destructive testing (NDT) method [124, 168]. It is a multi-angular analysis followed by a mathematical reconstruction that produces images of trans-axial planes through the body from a large number of projections. In medicine, these projections are generally acquired using a coupled detector/x-ray source that rotates around the patient. Different data acquisition systems are discussed in Section 3.2.1. Then, the reconstruction process produces a 2D slice (called tomograph) of x-ray attenuation coefficients. Figure 3.2 is an example of a CT image of a liver and kidneys. 3D datasets can be formed by stacking 2D tomographs (see Figure 3.7). More recently, tomography with a spatial resolution better than 20 µm, called micro-tomography, has emerged. This can be achieved using micro-focus x-ray tubes, however the best images are produced by parallel and monochromatic x-ray beams of synchrotrons. This generates 3D images of micro-structure without an *a priori* model; for instance the 3D microscopy of a sample, about 1 mm in diameter, of bone from the calcaneum region obtained by biopsy [38].

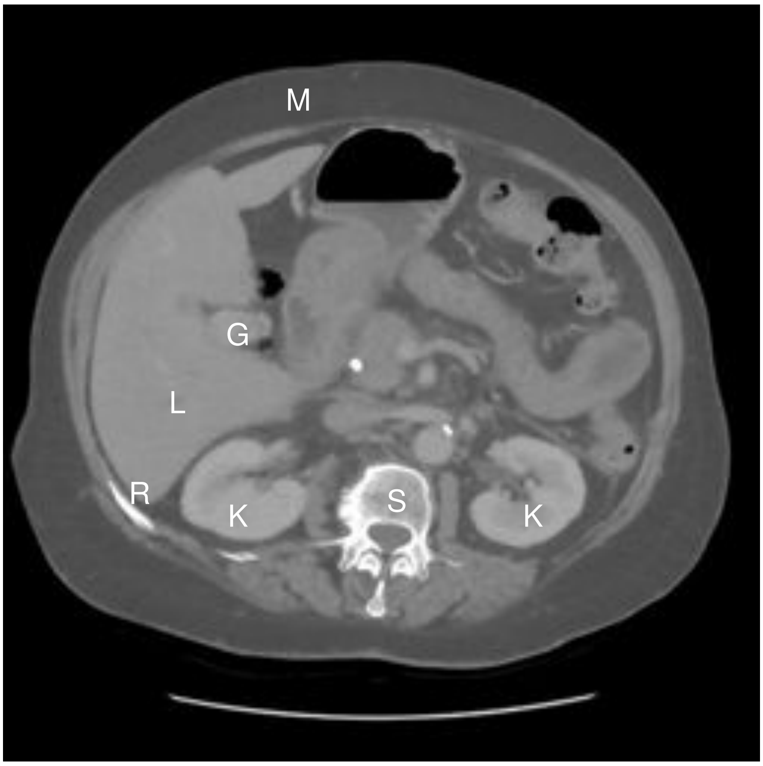


FIG. 3.2: *Example of CT slice, with G the gallbladder, K the kidneys, L the liver, M the muscle, R a rib, and S the spine. Image courtesy of Dr. Nicholas Chalmers at Manchester Royal Infirmary.*

Section 3.2.1 reviews the physical properties of x-rays and the mathematical concepts required for the physically-based simulation of fluoroscopic images from 3D CT data presented in Section 4.7.

3.1.2 Ultrasound Imaging

Independently, two decades after the discovery of x-rays, the first application of ultrasound in the field of medicine appeared. It was applied in therapy rather than in diagnosis. Diagnostic imaging techniques by 2D ultrasonography appeared in the early 1950's [278, 104, 100, 81]. The first systems acquired only single lines of data. Today it is possible to produce greyscale cross-sectional images using pulse-echo ultrasound in real-time. Pulses are generated by a transducer and sent into the patient's body where they produce echoes at organ boundaries and within tissues. Then, these echoes are returned to the transducer, where they are detected and displayed in real-time on a screen; Figure 3.3 is an example of ultrasound machines. Figure 3.4 is an ultrasonic image of a transverse liver. The biggest advantages of this technique are i) the fast acquisition time, which enables rapid diagnostics producing greyscale images of nearly all soft tissue structure in the body, and ii) unlike x-ray imaging techniques, US imaging techniques do not make used of ionazing radiations. Also, as Figure 3.3 illustrates, ultrasound machines can be portable due to their relatively small size. Two-dimensional ultrasound (2D US) has been used routinely in obstetrics and gynaecology since the 1960's [81].

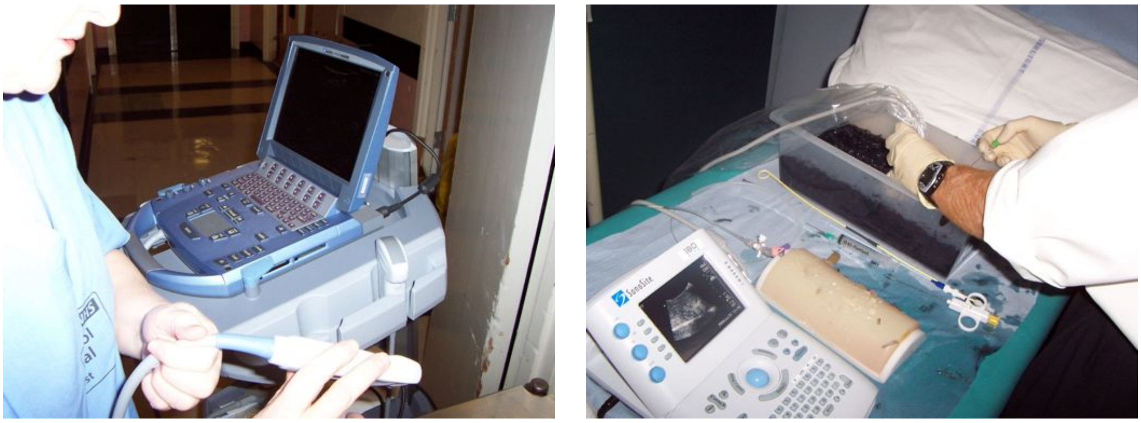

(a) (b)

FIG. 3.3: *Examples of US machines.*

Nevertheless, ultrasound is not restricted to the field of one or two dimensional (1D/2D) signals. Indeed 3D imaging by ultrasound has been available since 1974 [46] and clinically available since the 1990's. This method is well suited for examining the heart [46, 221, 71] as well as embryos and foetus [81]. The first techniques of 3D ultrasound were off-line and based on the combination of 2D images and their spatial position obtained with a mechanical articulated arm, into a 3D volume. A second method, called the freehand technique [13], is also used

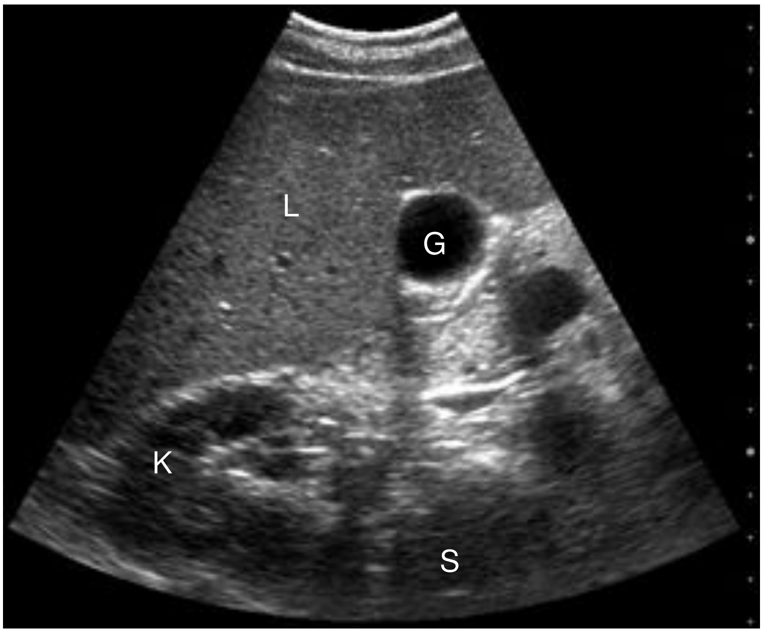


FIG. 3.4: *US image of a transverse liver, with G the gallbladder, K the kidney, L the liver, and S the spine. Image courtesy of Dr. Nicholas Chalmers at Manchester Royal Infirmary.*

to acquire volumes of ultrasound data. A position sensor is mounted on a conventional 2D US transducer. The transducer is swept through the region of interest and the position sensor allows the capture of multiple 2D US images. Several other data acquisition techniques are available [136]. By gating 3D ultrasound acquisition with the electrocardiogram (ECG) signal of the heart, 4D images of the beating heart can now be obtained.

Section 3.2.2 details the basics of ultrasound imaging needed for the approximation of US images from CT data presented in Section 4.6.

3.1.3 Nuclear Magnetic Resonance Imaging

Another fundamental discovery in the field of 3D medical imaging is the technique of nuclear magnetic resonance imaging (NMRI) [128], also called magnetic resonance imaging (MRI) due to the negative connotations associated with the term “nuclear”. It is also a tomographic technique, which produces the image of the NMRI signal in a slice through the human body. Figure 3.5 shows examples of MRI slices. According to current knowledge, MRI is totally harmless, contrary to x-ray tomography that uses ionizing radiations. Thus MRI has a great advantage over 3D x-ray imaging. However MRI is proscribed in patients with metal implants or pacemakers

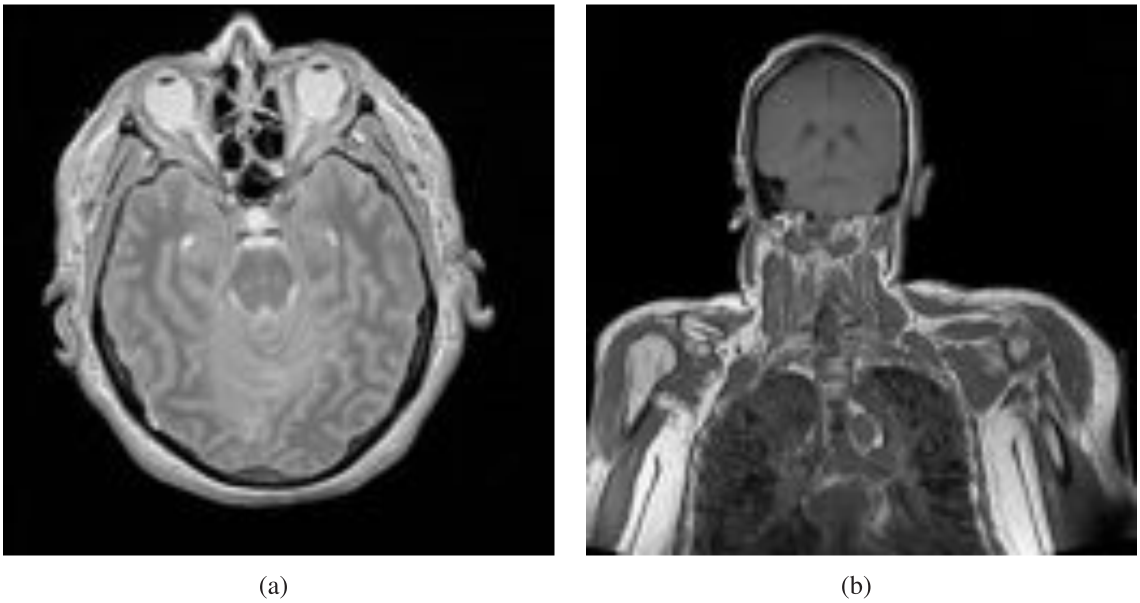

(a) (b)

FIG. 3.5: *MRI slices of a female cadaver collected for the National Library of Medicine's Visible Human Program [177].*

due to its use of a strong magnetic field.

The fundamental principle of the magnetic resonance phenomenon was discovered independently by Felix Bloch and Edward Purcell in 1946 (Nobel Prize in 1952). In a strong magnetic field, atomic nuclei rotate with a frequency that depends on the strength of the magnetic field. Their energy can be increased if they absorb radio waves with the same frequency (resonance). When the atomic nuclei return to their previous energy level, radio waves are emitted. Between 1950 and 1970, NMRI was developed and used for chemical and physical molecular analysis. In 1971, Raymond Damadian demonstrated that the nuclear magnetic relaxation times of tissues and tumours differed, consequently scientists were considering magnetic resonance for the detection of disease. In parallel to the development of x-ray tomography, Paul Lauterbur tested MRI tomography using a back projection algorithm similar to that used in x-ray tomography (see the paragraph on "X-ray Computerized Tomography" in Section 3.2.1). In 1975, Richard Ernst proposed MRI using phase and frequency encoding, and the Fourier Transform (Nobel Prize in Chemistry in 1991), which is still the base of current MRI. Two years later, Raymond Damadian demonstrated MRI of the whole body and Peter Mansfield developed the echo-planar imaging (EPI) techniques. Since then, MRI has represented a breakthrough in medical diagnostics and research, allowing human internal organs to be imaged with exact and non-invasive methods without depositing an energy dose. In 1993, functional magnetic resonance imaging (fMRI) was developed. This allows the mapping of the function of regions of the brain. Lauterbur and Mansfield won the Nobel Prize in Physiology or Medicine 2003 for their discoveries concerning magnetic resonance imaging.

3.2 Applied physics and mathematics

Section 3.2.1 summarises the theoretical principles underlying x-ray CT imaging. CT is the input data that is most commonly used in medical virtual environment (VE). This section also includes physical and mathematical concepts related to x-ray image techniques required for the understanding of Section 4.7 about the simulation of fluoroscopic images on the graphics processing unit (GPU). Section 3.2.2 provides the main concepts of ultrasound imaging techniques. It also describes images properties and artefacts characteristic of US images that need to be reproduced to generate US-like images from CT data (see Section 4.6).

3.2.1 X-ray Imaging and Computerized Tomography Principles

X photons are an electromagnetic radiation (like visible radiation), generally emitted when matter is bombarded with fast electrons (x-ray tubes). Their wavelengths are very small (typically from 10^{-10} to 10^{-14} metres).

Main Interactions of X-rays with Matter

X-photons cross matter. During their path into a sample or the human body, they can interact with matter according to various mechanisms [7]. There are four main kinds of interaction:

- **Photoelectric effect.** The energy of a photon is completely absorbed by an atom. Its energy is transferred to an electron of an internal layer (generally K or L). This electron is then ejected from its atomic orbital. The ionization of this electron layer is followed by the emission of a X-fluorescence photon or an Auger electron (probability of Auger effect is important for low atomic number). Generally, fluorescence photon is absorbed near the location of the interaction because of its low energy.
- **Rayleigh scattering (or coherent scattering).** As a first approximation, photons can be considered deviated from their direction without energy loss.

- **Compton effect (or incoherent scattering).** This effect is the result of an elastic shock between an electron and an incident photon. This electron is free or weakly linked to the atom. The photon transfers a part of its energy to the electron. The photon is scattered, and the electron gets kinetic energy.

- **Pairs creation.** This effect is based on the conversion of incident photon energy into mass energy. An electric field is around the nucleus of an atom (or around an electron). This field can transform the incident photon into matter. This matter corresponds to a pair of electron-positron. It only appears when incident energy is superior to 1.02 MeV (minimum energy corresponding to the mass creation).

Note that a high energy photon (from 6 to 15 MeV) can be absorbed by a nucleus that causes a nuclear reaction (the nucleus becomes unstable, it disintegrates and emits a proton or a neutron) but this does not happen in the case of x-ray imaging.

Figure 3.6 illustrates the image formation, and the interactions between photons and matter. For most x-ray imaging modalities, including fluoroscopy, only directly transmitted photons are essential, because they project on the detector an image of the sample, or of the human body. The quality of the image produced depends on the interactions of photons into the human body and into the detector. Indeed, as it can be deduced from Figure 3.6, photon scattering blurs the resulting image. Consequently, the hardware-accelerated implementation proposed in Section 4.7 to simulate fluoroscopic images by computing the x-ray attenuation, using the Beer-Lambert law (see below), through voxel data is restricted to directly transmitted photons.

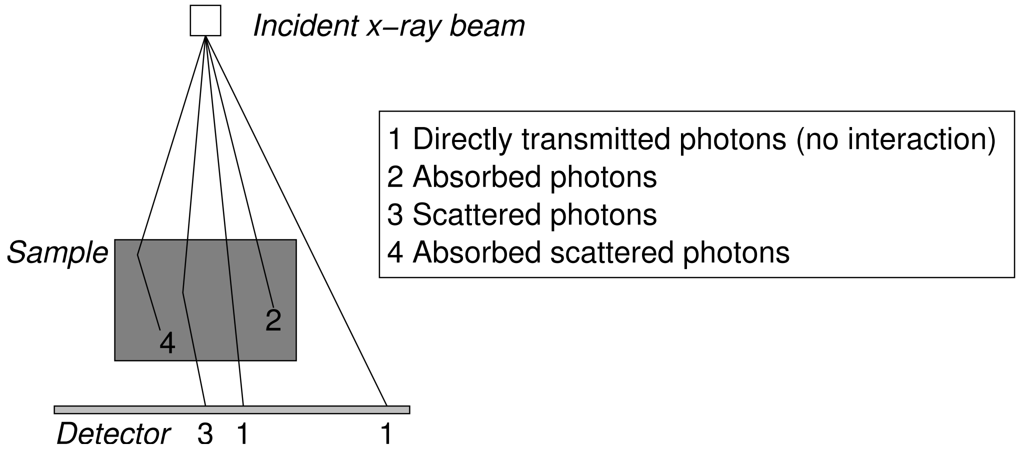

Incident x−ray beam
1 Directly transmitted photons (no interaction)
2 Absorbed photons
3 Scattered photons
4 Absorbed scattered photons
Sample
4
2
Detector
3
1
1

FIG. 3.6: *X-photons/matter interactions.*

Attenuation Law

The attenuation law, also called the Beer-Lambert law, relates the absorption of light to the properties of the material through which the light is travelling. Along a given x-ray path, let N be the number of photons at abscissa x. The number of photons dN, which will interact along an infinitesimal distance dx, is given by:

$$\mathrm{d}N = -\mu N \mathrm{d}x \tag{3.1}$$

Coefficient μ, which can be seen as a probability of interaction by unit length, is also called linear attenuation coefficient. It is usually expressed in cm^{-1}. It depends on i) E - the energy of incident photons, ii) ρ - the material density of the sample, and iii) Z - the material atomic number of the sample.

Eq. 3.1 can be integrated over the spatial and energy domain. Let N_{in} be the number of incident photons (or the input intensity). The total number of transmitted photons, N_{out} (or the output intensity), becomes:

$$\int N_{out}(E)\mathrm{d}E = \int N_{in}(E) \times \exp\left(-\int \mu(\rho(x), Z(x), E)\mathrm{d}x\right)\mathrm{d}E \tag{3.2}$$

This integrated form is the Beer-Lambert law. Simplified forms can be used, for example when the incident x-ray beam is monochromatic:

$$N_{out} = N_{in} \times \exp\left(-\int \mu(\rho(x), Z(x))\mathrm{d}x\right) \tag{3.3}$$

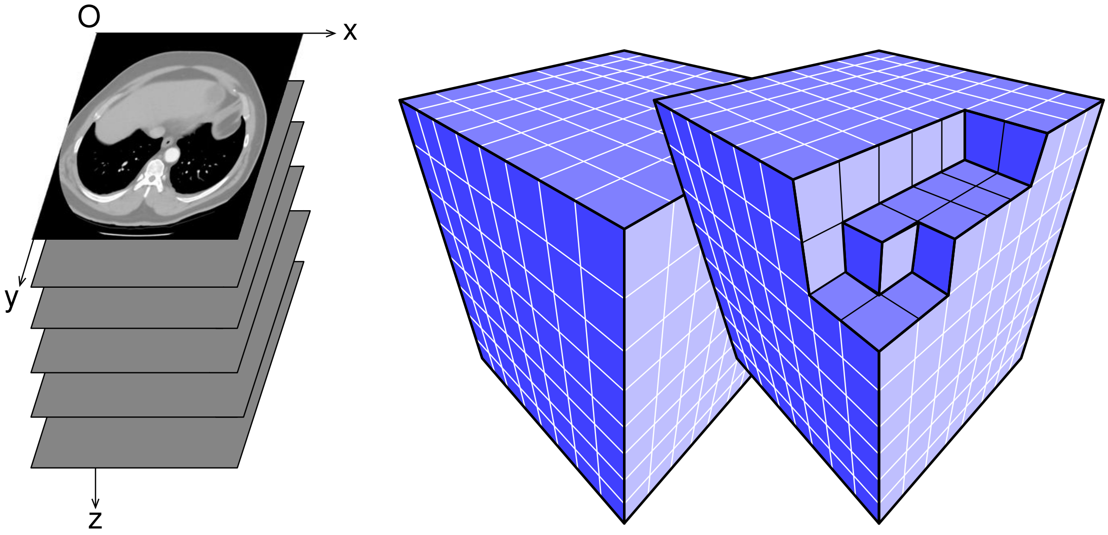
Furthermore, for homogeneous materials, we get

$$N_{out} = N_{in} \times e^{-\mu x} \tag{3.4}$$

Note that generally, μ is not directly used. The mass attenuation coefficient μ/ρ (in cm^2.g^{-1}) is often preferred. Indeed, it does not depend on the density and state of the material, *i.e.* at a given energy, μ/ρ is constant for a given chemical composition $\left(e.g.\ \left[\frac{\mu}{\rho}\right]_{ice} = \left[\frac{\mu}{\rho}\right]_{water} = \left[\frac{\mu}{\rho}\right]_{vapour} \ \&\ \rho_{ice} \neq \rho_{water} \neq \rho_{vapour}\right)$.

X-ray Computerized Tomography

Tomography, a NDT method, is a multi-angular analysis followed by a mathematical reconstruction. Today, x-ray computerized tomography is the commonest imaging modality in hospitals to routinely acquire three-dimensional (3D) data sets from patients. It produces a stack of trans-axial planes through the body, which can be considered as a 3D discrete volume dataset made of voxels (see Figure 3.7). In 1972, Hounsfield successfully tested the *EMI Mark I head scanner*, the first clinical computerized axial tomography (CAT) system [103]. Figure 3.8 is a photograph of a medical computerized tomography scanner currently in use at the Radiology Department of the Royal Liverpool University NHS Trust.



(a) Stack of 2D slices.

(b) Voxel dataset.

FIG. 3.7: *Discrete volume dataset.*

CT data acquisition consists of acquiring many projections at different angles. The scanning geometry has evolved since the first generation of medical CT systems. To date, we can distinguished five generations of scanners. Currently most clinical scanners are third or fourth generation. Their data acquisition processes are illustrated in Figure 3.9. They make use of a fan-beam x-ray source, so that the full width of the patient is covered by the incident beam. During the data acquisition, the patient is translated at a constant speed while the gantry rotates, this is called “helical” or “spiral” scanning. In the case of the third generation, between each projection, the couple source/arc detectors is rotated around the patient (see Figure 3.9(a)). In the case of the fourth generation, the detectors form a complete ring, then only the x-ray source

FIG. 3.8: *Medical CT-scanner: Siemens SOMATOM Sensation 16. Image courtesy of Dr Derek A. Gould at Radiology Department, Royal Liverpool Trust.*

is rotated between each projection (see Figure 3.9(b)). The next generation of scanners is now commercially available. Helical scanning is also used but these scanners make use of a multi-row detectors. More details about scanning geometries are available in [168, 230].

The reconstruction process requires back-projecting into the object space a physical variable whose measures are integral quantities along straight lines. From the Attenuation Law we know that an x-ray projection delivers at each pixel an integral measure of the attenuation coefficient. Therefore after reconstruction, a tomographic slice ideally corresponds to a cross-sectional map of linear attenuation coefficients (μ). Figure 3.10 illustrates in 2D the expression of a projection at angle θ that can be expressed as follows:

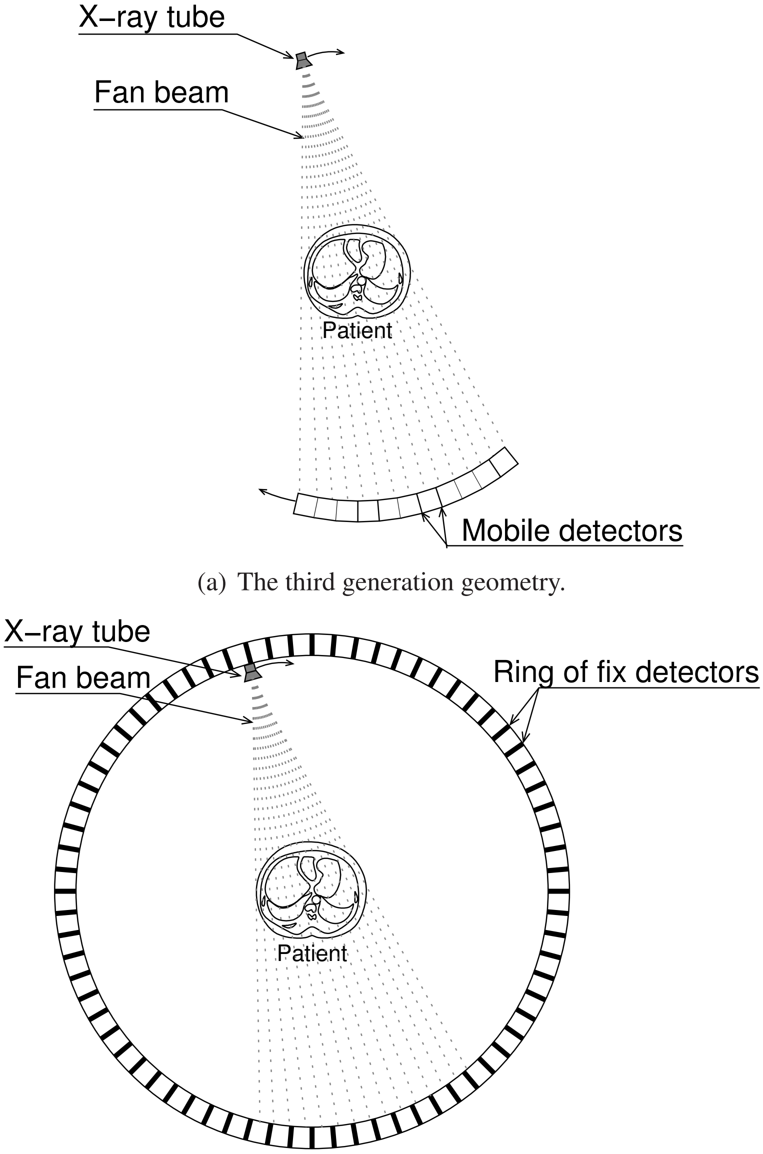
$$P_\theta(u) = \int\int f(x,y)\delta(u - x\cos\theta - y\sin\theta)\mathrm{d}x\mathrm{d}y \tag{3.5}$$

There is a relationship between the expression of a projection (Eq. 3.5) and the Radon transform (Eqs 3.6 and 3.7)

$$R_f(t,\theta) = P_\theta(t) \tag{3.6}$$

$$R_f(t,\theta) = \int\int f(x,y)\delta(t - x\cos\theta - y\sin\theta)\mathrm{d}x\mathrm{d}y \tag{3.7}$$

Figure 3.11 illustrates the process of the tomographic reconstruction. The sinogram, which is the observed measure, *i.e.* the experimental data, is built from the set of x-ray projections at succes-



(a) The third generation geometry.

(b) The fourth generation geometry.

FIG. 3.9: *Examples of X-ray computed tomography acquisition systems.*

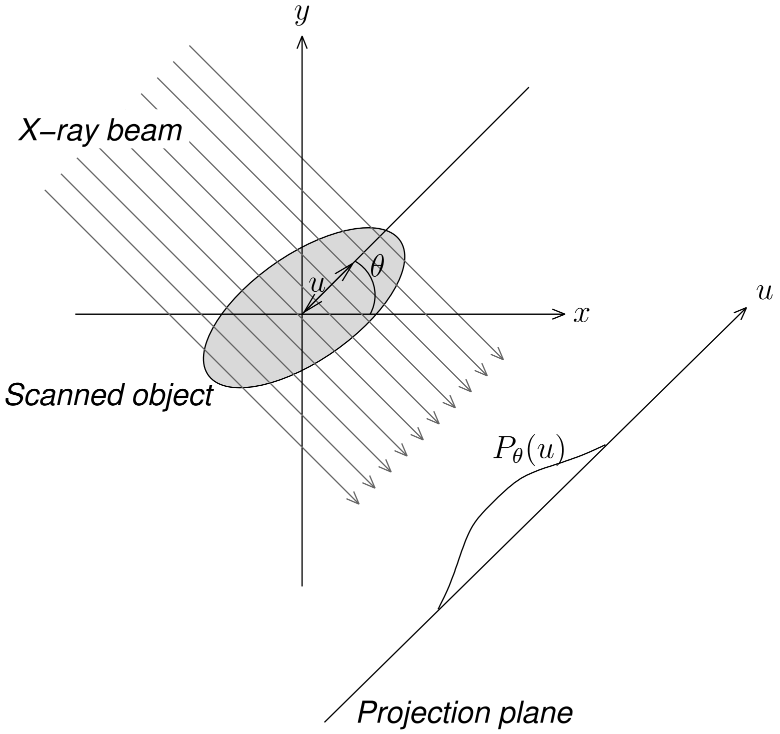


FIG. 3.10: *Projection expression.*

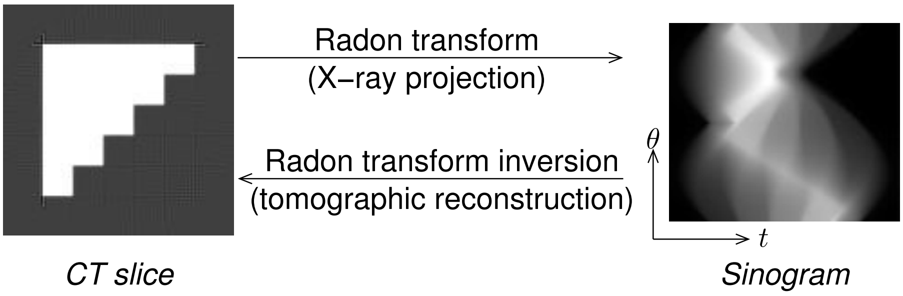


FIG. 3.11: *CT reconstruction of an unknown slice from a known sinogram.*

sive angles. Then, tomographic reconstruction is an inverse problem that consists of estimating the original object from its sinogram. There are two main classes of tomographic reconstruction: i) discrete reconstruction methods (also called algebraic reconstruction techniques), and ii) analytic reconstruction methods. The discrete case uses iterative correction algorithms. For example, one of the most commonly used algorithm consists of updating the reconstructed volume for each measured pixel location, *i.e.* ray by ray. In the analytic case, methods are based on continuous modelling. The reconstruction then consists of the inversion of measurement equations. The most frequently used technique is the filtered back-projection (FBP) algorithm. It inverses the Radon transform using the Fourier slice theorem:

Theorem 1: *the 1D Fourier transform of a projection is equal to a slice of the 2D Fourier transform of the original image.*

The complete 2D Fourier transform of the image is reconstructed from these 1D Fourier transforms. Then, the image is obtained by inverting its Fourier transform. More details about these two kinds of tomographic reconstruction algorithms are available in [124].

In medicine, images are commonly distributed over the hospital network using the Digital Imaging and COmmunications in Medicine (DICOM) standard. It includes a file format defined in the part 10 of the DICOM version 3 standard [176]. It was created in 1990, when there was no medical image file format standard. It responded to a need for an image storage and communication format. In practice, most manufacturers support the DICOM standard in their medical imaging systems. Although a x-ray CT slice typically corresponds to a cross-sectional map of linear attenuation coefficients at the energy of the incident beam used during the scanning of the patient, most of the medical scanners employ the $[-1000, 1000]$ scale. This standard scale is often called Hounsfield scale. A value H in Hounsfield units is given by:

$$H = 1000 \times \left(\frac{\mu}{\mu_w} - 1 \right) \tag{3.8}$$

with μ_w the linear attenuation coefficient of liquid water at a given energy. Then, H_{water} is 0, H_{air} is -1000 and H_{bone} is approximately 1000.

3.2.2 Ultrasonography

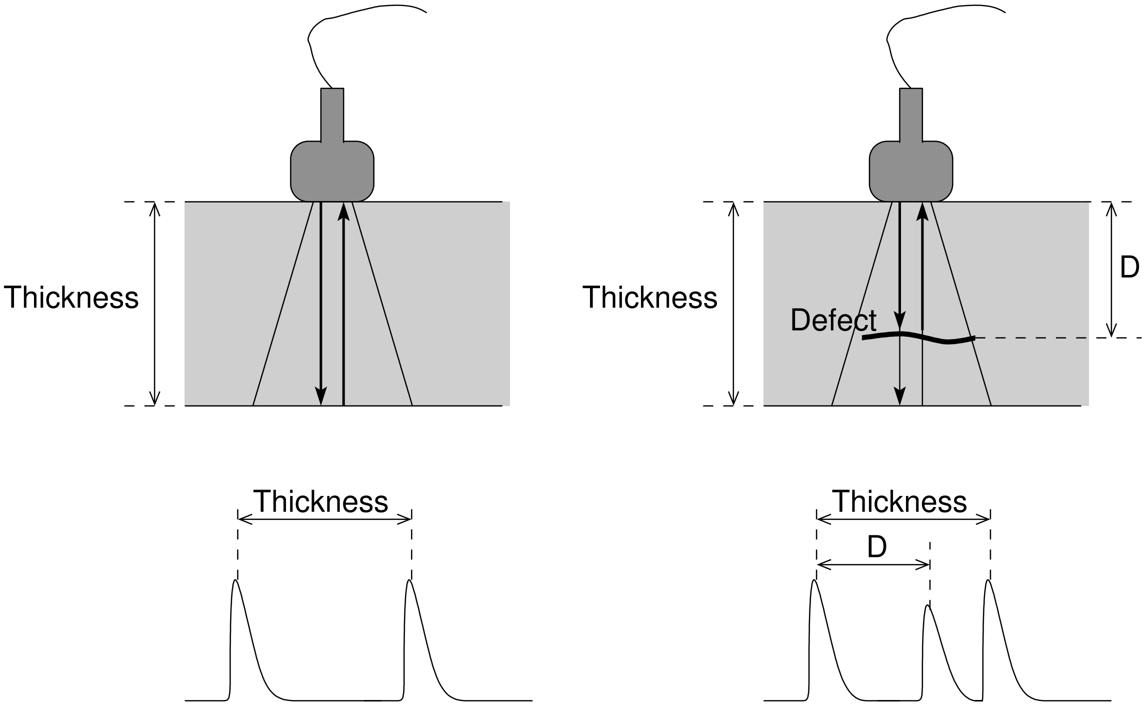
The frequency of audible sounds lies within the range 15-20 kHz. The term “ultrasound” comes from the fact that the frequency is too high to be audible by human beings. Then ultrasound is a sound whose frequency is higher than 20 kHz. Similarly infrasound is a sound whose frequency is below the audible range. For medical imaging, the frequency typically used is within the range 3-15 MHz.

Ultrasound Imaging Principles

The principle underlying ultrasound imaging, also called *ultrasonography* and *echography*, is the **pulse-echo**. In the medical context, it consists in emitting a ultrasound pulse using an emitter in contact with the patient's skin. The ultrasound waves will propagate through tissues. When a wave reaches the interface between two kind of tissues, *i.e.* their acoustic impedance is sufficiently different, there is reflection. Reflected waves, *i.e.* the **echo**, are detected by the receiver, which is typically housed in the same device as the emitter - such a device is called an ultrasound *transducer* or *probe*.

Pulse-Echo Method (A-scan) The earliest scanners displayed the signal on an oscilloscope as a single line corresponding to the amplitude of the incoming echo signals. This is the **A-mode** or **A-scan** method, as illustrated by Figure 3.12. It uses an ultrasound pulse and records the time variation of the echoes. As stated previously, reflections occur when an ultrasonic pulse hits the frontier between two mediums of different impedance. The delay between the transmission of the pulse and the reception of the echo depends on the velocity of the impulse and the length of the path to and from the reflecting surface. The signal observed is a series of deflections corresponding to the number of interfaces through which the waves have travelled. Obviously, when the ultrasound pulse goes through a medium without interfaces, no echoes can be returned.

Grey-scale Ultrasound Image (B-scan) A **B-scan** is a 2D signal made of a large number of adjacent 1D A-scans. Then, it is displayed as a grey-scale 2D image. The brightness of each pixel is determined by the echo amplitude (after time gain compensation (TGC)) so that interfaces are shown as bright dots instead of deflections from the baseline. However, the fundamental information-coding phenomenon in ultrasound imaging is **scattering**. The scattered ultrasonic wave contains information about the tissue structure [36]. Indeed, in the medical context, tissues can also be considered as a set of non-uniform small-scale structures. A major part of the grey-scale image is the result of backscattered waves from such small-scale structures that vary in size and shape, and are partly random in position and orientation. Using selective amplification of the low level echoes, backscattered waves are converted into a so-called **grey-scale ultrasound**



(a) Homogeneous material.

(b) Homogeneous material with a defect.

FIG. 3.12: *Pulse-echo method (A-scan).*

image. Then, the observed US images are degraded representations of the tissue reflectance. Note that at the frequencies used for diagnostics, ultrasound cannot penetrate air or bones. This results in shadowing.

The major advantages of ultrasound are: i) the fast acquisition time, which enables realtime diagnostics producing grey-scale images of nearly all soft tissue structures in the human body, ii) the flexibility of scanning, *e.g.* the operator can produce planes in longitudinal, oblique or any other plane according to the orientation of the transducer, and also portable US machines are available, and iii) unlike x-ray imaging techniques, ultrasound imaging techniques do not make use of ionizing radiations.

Phase Fluctuation Method (Doppler Method) The Doppler effect consists in a change in the frequency of a wave. It results from the motion of the wave source, the motion of the receiver, or the motion of the reflector in the case of a reflected wave. In medicine, Doppler ultrasound is used to detect and measure moving structures, *e.g.* blood flow [39]. A wave is transmitted into the tissue and the frequency of the echo is compared with the original. A rise in frequency indicates a movement toward the transducer, and *vice versa*. The frequency change, or Doppler shift, is proportional to the velocity of the moving object. Blood flow can be visualized in colour, *e.g.* red or blue, according to the direction of the flow in relation to the probe.

Terminology

Longitudinal waves In longitudinal waves, the oscillations of the medium, through which the waves are travelling, are parallel to the propagation direction of the waves. Consequently, sound waves are longitudinal waves transmitted by the mechanical vibration through a medium, fluid or solid. In diagnostic pulse/echo ultrasound imaging methods, longitudinal waves are usually produced and measured using a probe made of arrays of piezoelectric crystals.

Piezoelectric effect Ultrasound transducers are usually made of arrays of piezoelectric crystals. Basically, piezoelectric materials generate a voltage in response to applying mechanical stress, this is the **piezoelectric effect**. Similarly, the inverse of the piezoelectric effect is that mechanical distortions are observed when an external electric field is applied on the material [241, 106, 11]. The piezoelectric effect and its inverse are used to measure the echo and produce the pulse respectively. The ultrasound pulse can be generated by applying an electric voltage to piezoelectric crystals within the transducer that is directed toward the patient's skin. The echo is produced by reflected sound waves. The waves returning to the transducer deform the piezoelectric crystals, which will consequently produce an electric pulse (the echo).

Ultrasound pulse This is a short-duration wave of ultrasound, as opposed to continuous wave ultrasound. They are used in all ultrasound imaging modalities based on the pulse/echo methods where an ultrasound transducer transmits brief ultrasound pulses that propagate into the tissues, such as A-mode, B-mode, and the Doppler method .

Acoustic Impedance This is the property of medium causing resistance to the propagation of ultrasound. It only depends on the density of the medium. It is defined as follows:

$$Z = r \times C$$

with Z the acoustic impedance of the tissue, r its density, and C the velocity of the ultrasound wave in the tissue. It is measured in Rayls.

Velocity When an object is moving from Point A to Point B, one of the properties of its movement is the velocity, *i.e.* the ratio of the distance between these two points to the duration of the movement. Similarly, the velocity of a mechanical wave is constant in an uniform and homogeneous medium and is defined as follows:

$$C = \frac{d}{\tau}$$

with C the sound velocity, d the distance travelled by the sound wave and τ the duration of the movement. The velocity of sounds in each medium depends on the material properties (elasticity and density) of that medium. The velocity of sounds is nearly constant in the soft tissues of the body. For ultrasound machine set-up and calibration, it is assumed to be 1 540 metres per second. The velocity in air and bones differ greatly from soft tissues. The velocity of a wave can also be defined using its frequency and its wavelength:

$$C = f \times \lambda$$

with C the sound velocity, f its frequency and λ its wavelength. Note that during the propagation of the wave through different tissues, the frequency is constant. Therefore, the wavelength must change as the velocity in the medium changes. A sound wave can be considered as an ultrasonic wave if its frequency is higher than 20 kHz and if its wavelength is about 1.5 mm.

Ultrasound Images Properties

Tissue Interfaces This is the spatial location where a tissue of one type is adjacent to a tissue of another type. In this case, their acoustic impedance is sufficiently different. At tissue interfaces, ultrasound waves can be reflected, refracted or scattered.

Reflection and Refraction When an ultrasound wave hits an interface between two materials at an oblique angle, and the materials have different indices of refraction, it produces both reflected and refracted waves. Figure 3.13 illustrates the reflection and the refraction of a sound wave. Snell's Law describes the relationship between the angles and the velocities of the waves as follows:

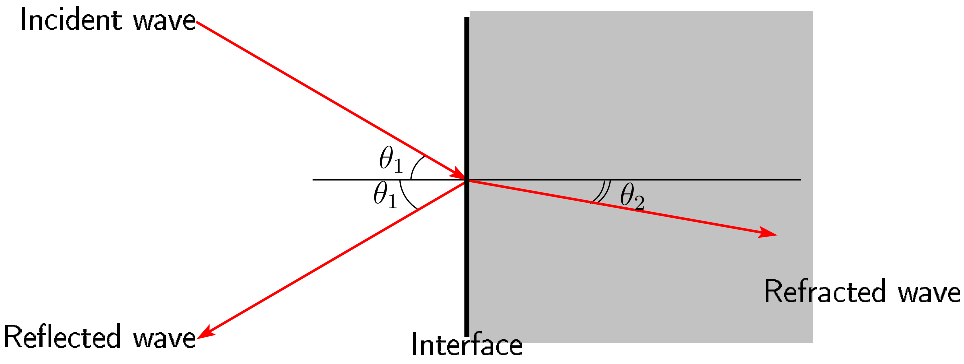


FIG. 3.13: *Reflection and refraction of a sound wave.*

$$\frac{\sin\theta_1}{C_1} = \frac{\sin\theta_2}{C_2}$$

with θ_1 and θ_2 the angle of reflection and the angle of refraction respectively; C_1 and C_2 the wave velocity in the medium 1 and 2 respectively.

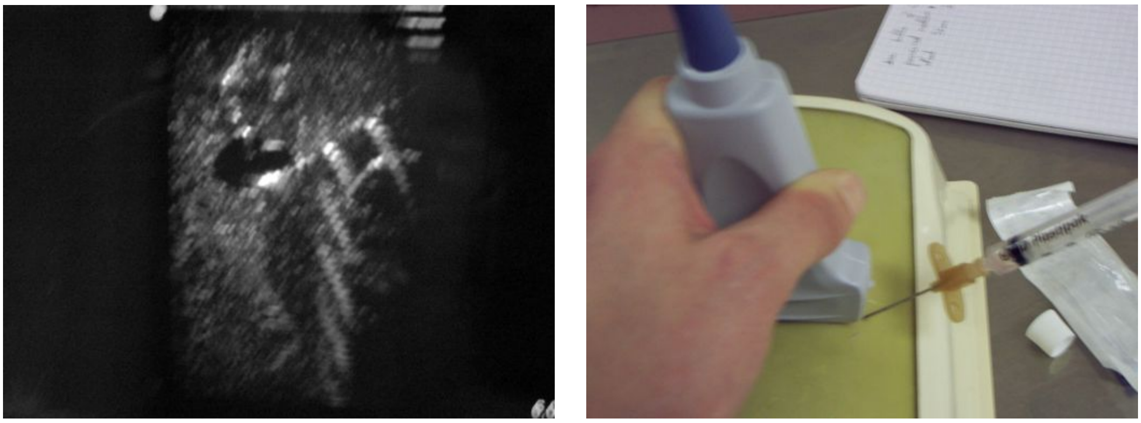
1. In a similar way that a mirror will reflect light, the reflection of an ultrasound wave occurs when this wave hits the interface between two mediums. Depending on the angle of reflection, the echo may return to the ultrasound transducer to provide a signal. The echoes are optimum when the sound waves are reflected back at 90°. However, reflections at this

angle only occur under special circumstances in medical ultrasound images. Reflections at other angles may contribute to a distorted image and/or artefacts.

Specular reflections may occur at the interface of a smooth large surface. In medical ultrasonic imaging, this is the case of reflections from organs, diaphragm, and the needle. In this case, the specular reflector reflects the ultrasound waves just as a mirror does in the case of visible light, *i.e.* the angle of reflection is equal to the angle of incidence.

When multiple reflections occur, there is reverberation. For example, this can happen between the walls of a needle or between two anatomic reflecting surfaces. Reverberation can appear as a series of closely spaced discrete echoes. Figure 3.14 shows such streaks caused by the needle. This artefact is commonly called a comet tail artefact or ring-down artefact.

2. Refraction happens when the transmitted beam at the interface between two mediums is deviated from the path of the incident beam. Refraction may also result in a distorted image and/or artefacts.



(a) Streaks artefacts due to the needle

(b) Experimental setting

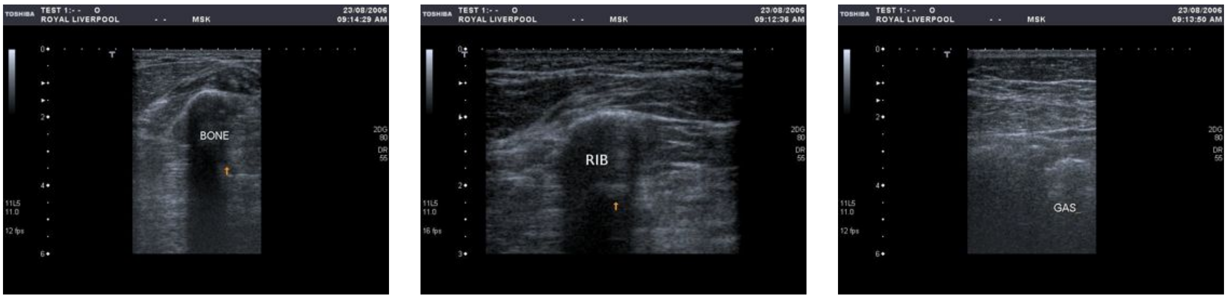
FIG. 3.14: *Reverberation artefact.*

Scattering This occurs when the sound wave hits a small irregular surface. This obstacle behaves as the secondary sound source and emits a wave, whose frequency is similar, in many different directions.

As previously stated in the paragraph on “Grey-scale Ultrasound Image (B-scan)”, scattering is the fundamental information-coding phenomenon in ultrasound imaging methods. In the medical context, tissues can be considered as a set of non-uniform small-scale structures of varying size and shape, and partly random in position and orientation that will backscatter waves. The scattered ultrasonic wave contains information about the tissue structure [36] and a major part of the grey-scale image results from such small-scale structures.

Attenuation and Absorption The amplitude of the sound decreases during its propagation through a medium. This is the attenuation phenomena. It is measured in dB.cm^{-1}. The attenuation of the signal can be caused by absorption, reflection or refraction.

Acoustic Shadowing Acoustic shadowing artefacts are typically observed in the area distal to a highly attenuating object that completely reflects and/or absorbs the incident ultrasound wave. In diagnostic medical ultrasound imaging, bones, calculi (solid masses) and gas are known to generate acoustic shadows. Indeed, no echoes are present in the area from the interface of such objects to the lower border of the image. Then this area is totally black, it is called hypoechoic area and corresponds roughly to the width of the interface. Figure 3.15 shows artefacts due to the bone of the arm, to a rib and to gas in bowels.



(a) Bone of the arm.

(b) Rib.

(c) Gas in bowels.

FIG. 3.15: *Acoustic shadowing artefact. Images courtesy of Dr Derek A. Gould at Radiology Department, Royal Liverpool Trust.*

Enhancement This effect appears at the echoes of interfaces located behind or within a low attenuation material, such as fluid. For instance, the vessel walls are enhanced due to the blood. Also, the area deep to fluid can appear brighter. Figure 3.16 is a longitudinal slice of the neck illustrating enhancement due to the jugular vein.

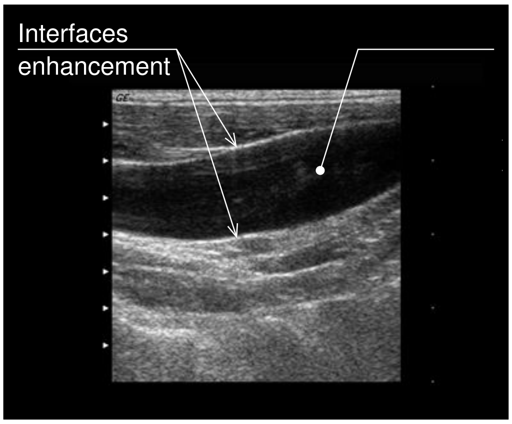


FIG. 3.16: *Example of a longitudinal slice of the neck showing the jugular vein illustrating enhancement in ultrasound images. Image courtesy of Dr. Nicholas Chalmers at Manchester Royal Infirmary.*

3.3 Algorithms and Techniques

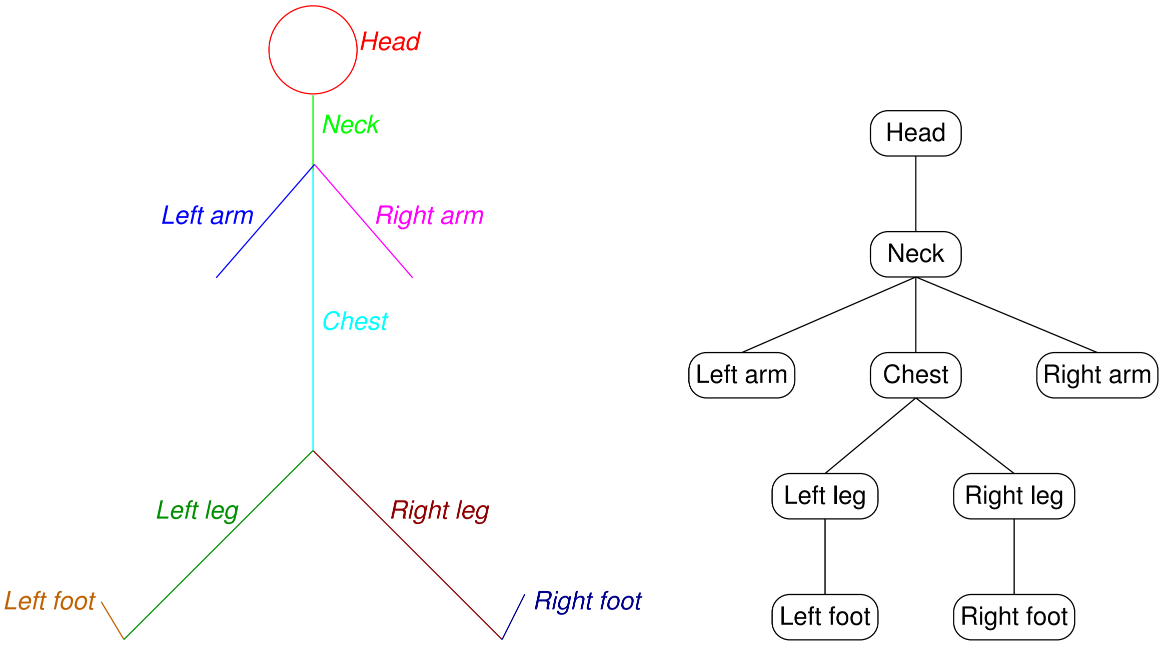
This section reviews some of the algorithms and techniques for medical visualization and haptics, with a focus on the methods used in medical training simulators. The section is divided into seven parts. First, it describes the notion of a "scene graph" used for the 3D scene management in graphics and haptic applications. Secondly it describes the filtering and segmentation steps that act as a pre-processing of medical image data before application of a visualization technique. Section 3.3.3 discusses the main approaches to the visualization of volumetric data in medicine. Brodlie *et al* [28] provides a more detailed survey. Section 3.3.4 briefly introduces the concept of "ray-tracing" needed for the understanding of Sections 4.7 and 5.3. More details are available in a new book on ray-tracing rendering [253]. Section 3.3.5 introduces GPU programming and general-purpose computing on graphics processing units (GPGPU). Common haptic rendering models are presented in Section 3.3.6. Sections 3.3.7 and 3.3.8 respectively present algorithms in use for the realistic physically-based simulation of x-ray and ultrasound images.

3.3.1 Scene graphs

Definition

A computer program uses its own algorithms and internal states to update a 3D scene and display a sequence of graphics and/or compute the haptic rendering. Within the program, this scene can be considered as a database, *i.e.* a collection of geometric objects, textures, lights, states, haptic properties, *etc.* During the rendering stage, queries will be performed, for instance to access visible objects and call their corresponding display functions. The database can also be interactively updated because of user interactions, *e.g.* via a graphical user interface (GUI), the keyboard, the mouse or a haptic device.

An efficient way to draw the scene is to organize the database as a graph, the *scene graph* (also written scene-graph). It is a data structure holding the information that represents the scene. Instead of simply containing a linear list of OpenGL [182] or DirectX [169] commands, the nodes of a scene graph can hold whatever data and procedures that are needed to manipulate and to expose the graphical and/or haptic scene to the user. This can be easily achieved because a scene graph is generally an object-oriented structure. The scene is represented using a hierarchical structure, such as a tree structure. Figure 3.17 is an example of scene graph for a humanoid object. A node may have many children but only a single parent. Thus, a parent is apparent to all its child nodes. If needed, an operation applied to a parent node can automatically propagate its effect to all the children. In many implementations, a geometrical transformation matrix is associated to each node of the scene graph.



(a) Schema of a simple humanoid.

(b) Corresponding scene graph.

FIG. 3.17: *A simple scene graph.*

The application starts by generating the scene graph (for an example, see Figure 3.17(b)). Then, during the rendering stage, the scene graph is efficiently traversed using an hierarchical order (see Figure 3.18).

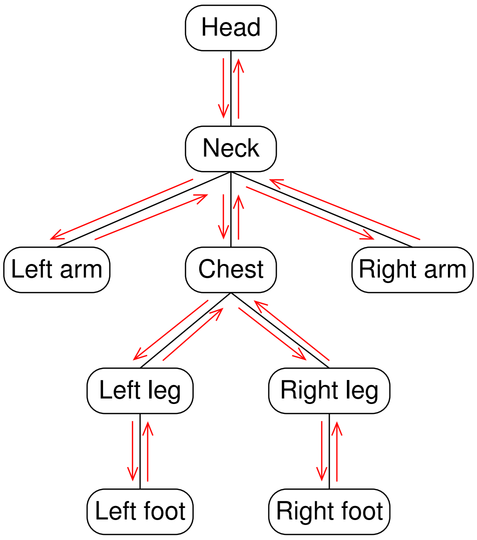


FIG. 3.18: *Scene graph traversal.*

In a medical virtual environment such as BIGN*e*PS*i*, a scene graph implementation is often used. This facilitates the traversal of the 3D scene during the haptic rendering and the graphic rendering. For instance, it makes possible to easily convert coordinates into local or global coordinates using two geometrical transformation matrices for each node. This is particularly useful to compute, using the global coordinates of the haptic devices, the response forces applied on these devices into the local coordinate system of the virtual patient, then to convert these forces into a coordinate system suitable for the haptic devices.

Development APIs

The following projects are the main application programming interfaces (APIs) for developers creating real-time visual simulations (in arbitrary order):

- Open Inventor [239], formerly known as IRIS Inventor [211];
- OpenGL Performer ™ [240], formerly known as IRIS Performer;
- OpenSceneGraph [184];

- OpenSG [183];
- OpenRM Scene Graph [203].

For more details about these implementations, the reader may visit the appropriate websites.

Web3D and exchange file formats

X3D [275] and its predecessor the Virtual Reality Modeling Language (VRML) [4] are both ISO ratified, royalty-free, and open standards file format designed particularly with the World Wide Web in mind to represent and communicate (interactive) 3D scenes and objects. X3D is based on extensible markup language (XML) [271], whereas the VRML syntax is rather similar to the HyperText Markup Language (HTML) syntax. MedX3D, an extension for X3D, is being designed for medical applications [274]. Similarly, COLLAborative Design Activity (COLLADA) [129] is an open file format based on XML. It was initiated by Sony Computer Entertainment to provide a standard format for exchanging data between multiple software packages from various vendors. One of the main advantages of COLLADA over X3D is its direct support from some of the major vendors in the industry of digital content creation, such as Alias, Discreet, and Softimage.

Haptics

To date, the following scene graph implementations have been designed to support haptics (in arbitrary order):

1. ReachIn API [206] is a proprietary implementation of a VRML scene graph;
2. H3D API [237] is an open source implementation of a X3D scene graph based on OpenHaptics [114];
3. Computer Haptics & Active Interfaces (CHAI 3D) [40] is a C++ oriented open source implementation of a scene graph;
4. OpenSceneGraph Haptic Library (osgHaptics) [269] is a C++ oriented open source scene graph based on OpenHaptics.

Scene-graphes based on Web3D standards are well designed for the rapid prototyping of haptic applications, whereas C++ oriented scene-graphes provide a relatively easier access to low-level functionalities.

ReachIn API The Reachin API [205]is a proprietary library developed by Reachin Technologies AB [206]. It implements a scene graph based on the VRML file format. The current version to date is 4.2. An application written with this makes use of C++, VRML and Python for the development. C++ [252] can be used for tasks requiring optimisations of computational time, creating new nodes, *etc.*; Python [202] for scripting, *i.e.* for all non critical tasks such as routing GUI events; VRML [4] for the scene graph description and also storing 3D geometric data. Features of the API include shader development using OpenGL ® Shading Language (GLSL) [213] (see Section 3.3.5), support of 3D surround sound system, support for 3DConnexion [1] devices, support of haptic devices by SensAble [233] and Force Dimension [68], *etc.*

H3D The H3D API [237] is an OpenSource library developed by SenseGraphics AB [238]. It implements a scene graph based on the X3D file format [275]. Support of a more up-to-date file format is obviously in favour of this API. The fact that it is also OpenSource is positive, moreover this gives support from the "community" as well as from the original developers of the API. Application written with this API makes use of C++, X3D and Python for the development. C++ can be used for tasks requiring optimisations of computational time, creating new nodes, *etc.*; Python for scripting, *i.e.* for all non critical tasks such as routing GUI events; X3D [275] for the scene graph description and also storing 3D geometric data. Features of the API include shader development using GLSL (see Section 3.3.5) or the CG toolkit [64], support of 3D surround sound system using OpenAL [97], support for 3DConnexion devices, support of haptic devices by SensAble and Force Dimension, but also all the devices supported by the Haptik Library [90].

Note that collision detection and haptic rendering are not currently a part of the H3D implementation. Force feedback computation, collision detection, haptic rendering and management of the haptic thread are performed using the Haptic Library (usually called HL) of the OpenHaptics toolkit (a proprietary API by SensAble), *i.e.* force feedback algorithms are only supported for PHANToM haptic devices by SensAble. For other devices, the developer needs to provide these functionalities.

The HaptikLibrary is only used for access to haptic devices. It supports various devices, including commercial devices by SensAble or Force Dimension. Haptic rendering and collision detection are not implemented since the library is only dedicated to the communication between the application and the device.

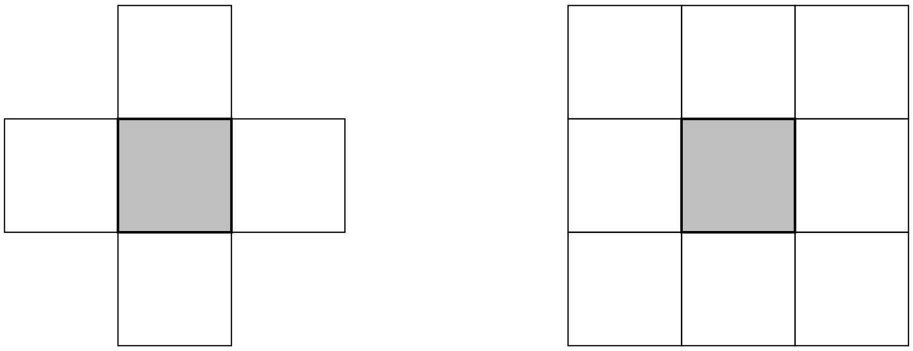
CHAI 3D The CHAI 3D API [40] is an OpenSource library developed by members of the Stanford Artificial Intelligence Laboratory (SAIL or Stanford AI Lab) [249]. It differs from the two previous API because CHAI 3D is “C++” oriented, which makes its utilization very flexible and the developer can have access to “low level programming” via C++. However, it also provides a similar command-line mesh viewer to load 3DS and OBJ files, which are commonly use in the games and computer graphics industry. In contrast to H3D, which is based on the HL library of the OpenHaptics toolkit, CHAI 3D implements its own collision detection algorithms, different haptic rendering techniques, *etc.* It supports force shading, mesh-to-mesh collisions, implicit functions, shader development using GLSL (see Section 3.3.5), support of sound system, *etc.*

OpenSceneGraph Haptic Library (osgHaptics) OpenSceneGraph Haptic Library (osgHaptics) [269] is an OpenSource library developed at VRlab, Umeå University [270]. It is C++ oriented, and is build on the top of OpenSceneGraph [184] and OpenHaptics. Features includes various features provided by OpenSceneGraph such as shader development using GLSL, various file formats, *etc.*

3.3.2 Image Filtering and Segmentation

One aim of image processing is to improve the quality of images. This can be used for example to facilitate their visual interpretation, to ease the manual or automatic extraction of relevant information, or to optimally represent the information contained in the image data.

Image data acquired by an imaging system always contains noise. Many techniques have been proposed in the past to reduce the noise level. This is typically done by smoothing the value of each pixel by some averaging over a local neighbourhood, *e.g.* using a *low-pass convolution filter*. In 2D a pixel can have 4 or 8 neighbours, these correspond to a neighbourhood in 4-connectivity or in 8-connectivity respectively (see Figure 3.19). The main drawback of this technique is that low-pass filtering reduces high frequencies of the image, in other words it blurs the boundaries between regions. A better approach for medical data is to use techniques based on anisotropic diffusion. These are non-linear filters where the image intensity values iterate



(a) Neighbourhood in 4-connectivity.

(b) Neighbourhood in 8-connectivity.

FIG. 3.19: *Pixel neighbourhood.*

towards an equilibrium state. The diffusion function depends on the magnitude of the gradient of the intensity. Diffusion occurs within regions where the gradient is small, and not across region boundaries where the gradient magnitude is high. This approach was first proposed by Perona and Malik [190] and it is now widely used.

In medical image processing, the next step after reducing noise is to identify different parts of the anatomy of particular interest. This is the so-called "segmentation". This will label pixels into contiguous regions with an identifier indicating the type of material. This process is often semi-automatic with user guidance to help correct identification. Segmentation is usually a major bottleneck in medical applications, it takes a long time and the results are often hard to reproduce because of the user involvement. Often the data is over- or under-segmented. Segmentation is a major field of study, supported by a considerable body of literature. This topic is generally covered in most books about image processing [82, 281]. A useful survey paper can also be found in [195]. We only give a very brief overview here. Segmentation algorithms are usually based on image properties such as discontinuities of the intensity values (*i.e.* the detection of abrupt changes such as edges in the image) or similarities according to a set of predefined criteria.

A typical strategy is to try simple techniques first, and seek a more sophisticated technique if these fail. The simplest technique is undoubtedly *thresholding*. This creates binary images according to the pixel intensities. All pixels with intensities greater than a given threshold are grouped in one class and all other pixels in another.

Another fundamental approach is *region growing* [10]. This technique is based on the spatial coherence of the data, *i.e.* connected pixels that succeed an alikeness criteria are in the same region. Usually a 3D seed point is manually selected in the volume dataset. An automatic algorithm “grows” the relevant region up to its boundary using a criteria of alikeness of neighbouring pixels. A more sophisticated class of region growing techniques makes use of *Markov random field models* [37, 107, 95, 283]. These are based on a probabilistic model that use the correlation between pixels in a neighbour. In practice, we compute the probability that a pixel belongs to a region using its own value and the values of its neighbouring pixels.

Another sophisticated segmentation technique is based on the *watershed transformation*. This is a mathematical morphology-based partitioning tool. An accurate description of the algorithm can be found in [210]. Imagine a landscape flooded by water. Watersheds are the dividing lines between basins, *i.e.* regions in image segmentation. The high dividing lines are abrupt changes in image intensities. This corresponds to a high magnitude of the gradient of the pixel intensities.

A further class of methods involve the user in guiding the segmentation process. In the *LiveWire segmentation* approach, the user selects, by cursor position, an initial point on the boundary. As the cursor is moved indicating a new point on the boundary, the algorithm dynamically computes an optimal path from the initial point to the current point and this is displayed in realtime [62].

Level-set segmentation is another kind of class of methods. These rely on partial differential equations (PDEs) to deform isosurfaces [185, 146].

Another class of segmentation methods are based on *deformable models*. These are dynamic models whose shape evolve under the influence of internal and external forces to fit as closely as possible a feature in the image [162].

In this thesis, we will use the anisotropic diffusion filter to reduce the noise level into the input data. To segment skin and bones from X-ray CT data, the thresholding method has proved to be efficient and this is the technique that we will use in this research. Section 4.2 provides details about the input data pre-processing and the generation of patient specific models that have been used in this thesis.

3.3.3 Volume Visualization

Isosurfacing

The principle of surface extraction algorithms is to extract explicitly an intermediate representation that approximates the surface of relevant objects from the volume data set. This is often called “isosurfacing’.’ For example, we might want to extract the surface of bones from a CT dataset. The surface extracted corresponds to a specified threshold value. The surface is extracted as a polygonal mesh, typically a triangular mesh, which can be rendered efficiently by graphics hardware (Figure 3.20).

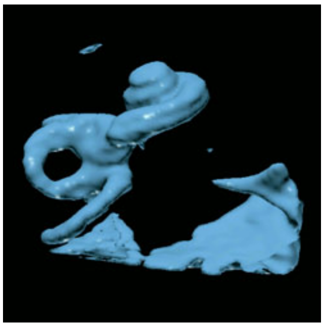


FIG. 3.20: *Isosurface of a human cochlea generated from MRI data. Image courtesy of Paul Hanns, University of Manchester.*

The most popular algorithm in use is Marching Cubes [156], or one of its derivatives [50, 88, 155]. An exhaustive review on Marching cubes algorithms can be found in [178]. The idea is to process the data cell-by-cell, identifying whether a piece of the surface lies within the cell, and if so, constructing an approximation to that surface. For large datasets, it is important to be able to identify quickly cells, which contribute to the surface: current research on fast identification of isosurface location can be found in [31] (contour trees) and [21] (searching “interval” space defined by maxima and minima of cells). Goetz *et al.* proposed a GPU implementation of the Marching Cubes algorithm but without span space or correct normals [80]. This is addressed in a GPU implementation of the Marching Cubes algorithm by Johansson and Carr [119].

Medical applications have been the first to exploit this facility. According to Schreyer and Warfield, most clinical applications involving 3D visualization can take advantage of this technique [227]. As an example, Figure 3.21 shows such a surface generated by *endovascular surgical planning software* implemented at the Manchester Visualization Centre (MVC) [191]. Here the high contrast in the source magnetic resonance angiography data make surface rendering a good technique for representing the network of blood vessels in the brain. In general, however, care must be taken to avoid the effects of false positives and false negatives as the surface extracted at a particular threshold value may not represent a true surface.

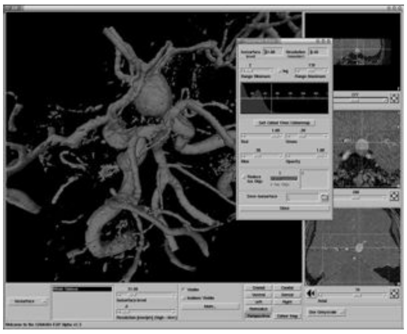


FIG. 3.21: *Surface extraction from MRA data of a patient suffering from a brain haemorrhage.* Source*: Reproduced from [120], published by the British Institute of Radiology.*

Volume Slicing

Of course a simple approach to visualizing a volume is to visualize a series of slices, either parallel to one of the faces of the volume, or in an oblique direction (see Figure 3.22). This is often called multiplanar reconstruction (MPR) and is arguably the most common visualization technique in clinical practice [65]. Radiologists are trained to move through the slices and to recognise branching patterns through this process. Their experience in moving through 2D slices in this manner allows them to build a 3D mental model of the real anatomy. A difficulty with this approach is that branching structures of interest such as blood vessels are non-planar and therefore cannot be easily followed in such an approach. A recent idea is to use curved planar reconstruction (CPR), where a “curved” slice following the trajectory of a vessel is presented [125]. Note however that this requires prior identification of the centreline of the vessel and so significant effort is required to create CPR visualizations.

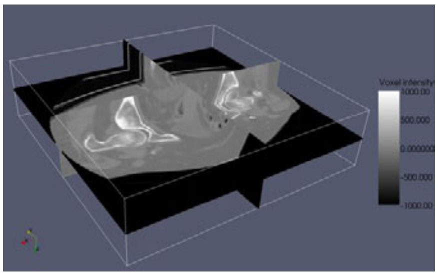


FIG. 3.22: *Image slicing using Paraview [131].*

Volume Rendering

While MPR is commonly used in practice, there are situations where a 3D view will give the radiologist valuable insight - for example, when patients have an unusual or complex anatomy or pathology. Volume rendering has sparked a very active field of research among computer scientists, to develop fast and effective ways of presenting 3D medical visualizations. This is the subject of the remainder of this section, which assumes the data are in the form of a 3D volume, and more precisely as a rectilinear grid of voxels. In (direct) volume rendering, as opposed to explicit surface extraction, voxel-based data are directly visualized without extracting an intermediate surface representation.

Each voxel of a 3D dataset is assigned a colour value and an opacity value, creating an entity that can be thought of as a multi-coloured gel. Voxels are assigned colour and opacity using the idea of a transfer function. A simple transfer function will assign colour and opacity from the value of the data, but in practice multi-dimensional transfer functions are used. For example, in the early work on volume rendering, the transfer function depended on both value and gradient, in order to highlight boundary surfaces [150]. More recently, curvature-based transfer functions have been proposed [130].

There are four basic approaches to direct volume rendering:

Ray casting creates a high quality image, at the cost of compute time, by casting a ray for each pixel into the volume (see Figure 3.23), and compositing the light reflected back to the viewer from a set of samples along the ray, taking into account the colour and opacity at each sample. Computations are performed from the image space to the volume dataset, thus it is called a backward projection method. The seminal paper is that of Levoy [149]; a modern reference indicative of the state of the art in ray casting is the paper by Grimm *et al.* [87]. An active research area is the use of off-the-shelf graphics cards for real-time volume rendering using ray-casting [201, 137, 60, 224].

Figure 3.24 shows another example from MVC's *endovascular surgical planning software*, illustrating the view of an aneurysm and surrounding blood vessels rendered by ray-casting [150]. Whereas Figure 3.27 is a volume rendering of a high resolution CT data set of a patient with a large tumour below the stomach.

Splatting creates a high quality image, at the cost of compute time. Here, it estimates the contribution of each voxel in the volume dataset to the final image. Voxels are “thrown”

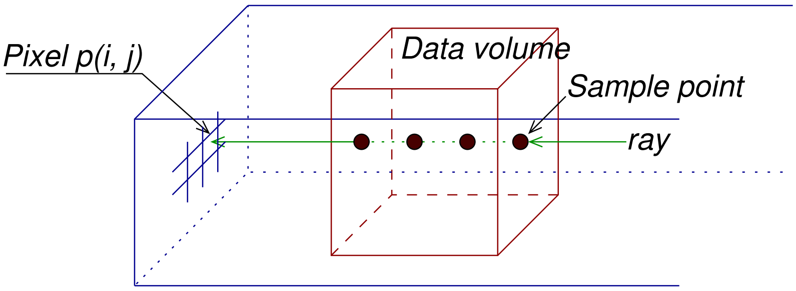


FIG. 3.23: *Ray casting volume rendering principle.*

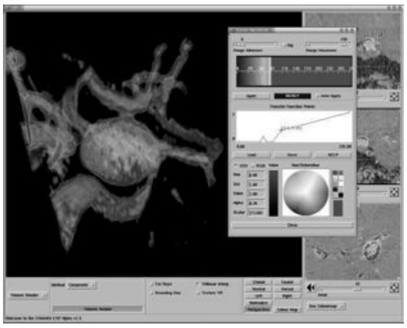


FIG. 3.24: *Ray-casting volume rendering from magnetic resonance angiography (MRA) data of a patient suffering from a brain haemorrhage.* Source*: Reproduced from [120], published by the British Institute of Radiology.*

at the image in a forward projection, forming a footprint, and the result is accumulated in the image plane (see [277] for original paper, and [105] for more recent algorithmic efficiencies).

Shear-warp provides lower image quality by speeding-up the rendering process. It is achieved by first aligning (using a shear) the volume and the viewing direction so that a line of voxels can project directly to a pixel, and secondly compensating for the first transformation by an image warp transformation (see Figure 3.25) (see [30, 139] for original paper and [255] for a recent reworking).

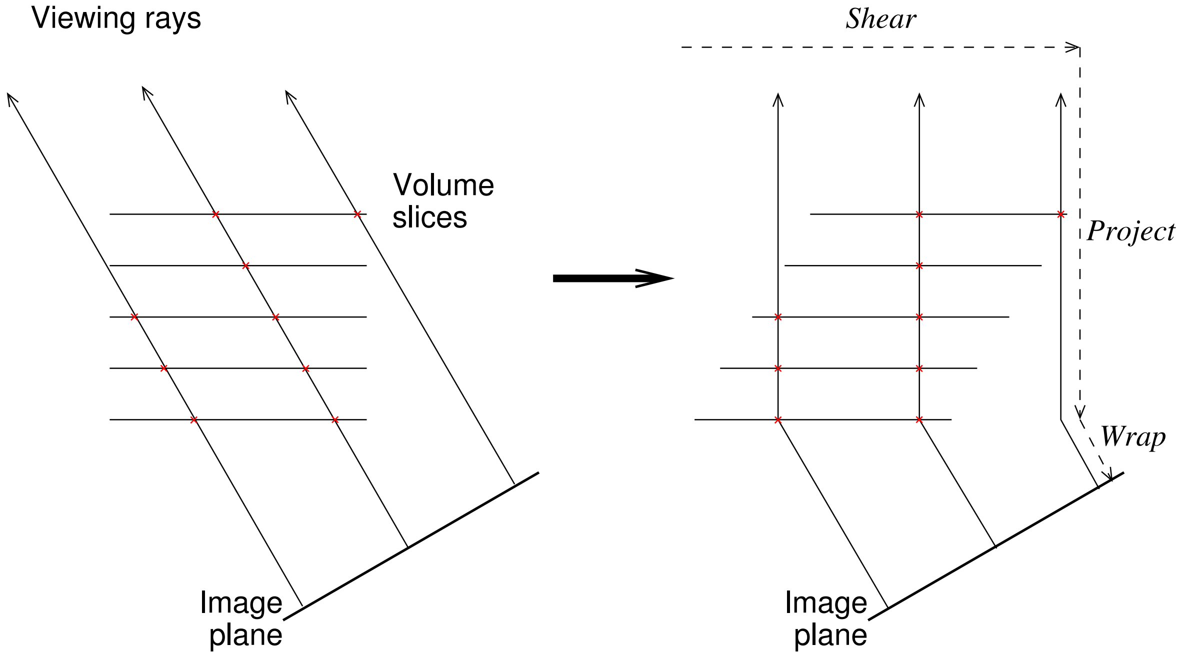
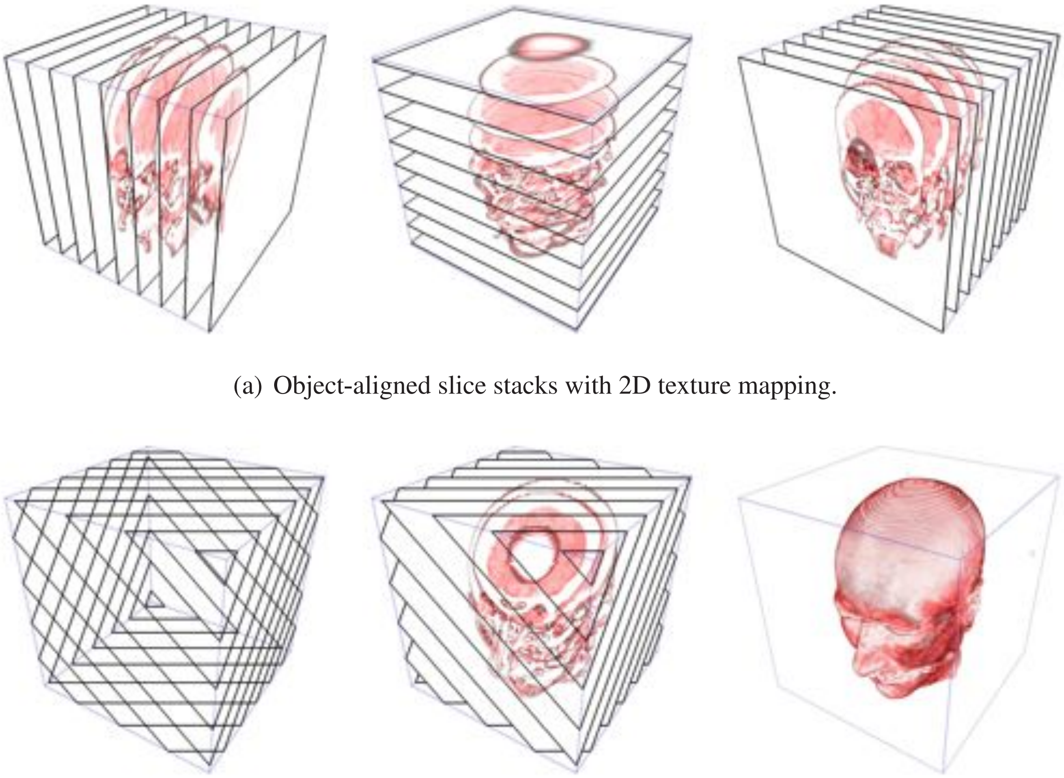


FIG. 3.25: *Shear-warp volume rendering principle for orthogonal projection.*

Texture-based methods consider the volume as three different stacks of aligned 2D textures (see Figure 3.26(a)) or as a single 3D texture (3.26(b)). Slices are projected and composited using graphics hardware (see Figure 3.26) (see [42] for initial ideas on texture-based volume rendering). The increasing power of graphics hardware is a major influence on algorithm design - see, for example, combined ray casting and textures [137]. A detailed review of this area (including a section on dealing with large data volumes) is contained in the SIGGRAPH tutorial [60].

Figure 3.27 shows an example of a 3D texture-based volume rendering obtained from CT data.



(a) Object-aligned slice stacks with 2D texture mapping.

(b) View-aligned slice stacks with 3D texture mapping.

FIG. 3.26: *Volume rendering using texture hardware.* Source*: Reproduced from [15], published by Springer Berlin / Heidelberg.*

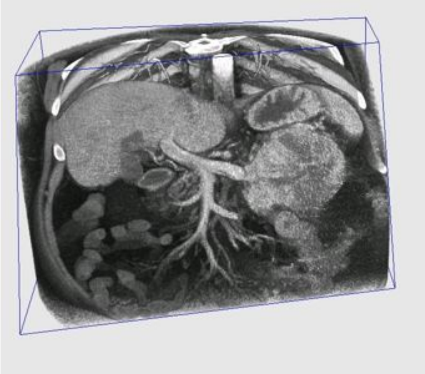


FIG. 3.27: *Volume Rendering generated from CT Data Scan showing the relationship between the confluence of the superior mesenteric, splenic and portal veins with a large tumour.*

A variation on volume rendering has proved successful in medical visualization, particularly for the visualization of blood vessels from MRI. This is maximum intensity projection (MIP). It uses images obtained after a contrast agent is introduced into the blood stream. The data values of vascular structures are then higher than surrounding tissue. Such examinations are referred to as either CT or MR angiography. By modifying the ray casting technique to pick out the voxel of maximum intensity (rather than compositing contributions from all voxels as would normally be done), we obtain a fast, effective and robust technique [174, 171]. MIP does have some drawbacks however. As the image contains no shading information, depth and occlusion information is missing. Small vessels tend to disappear in front of large bright vessels, due to the partial volume effect mentioned earlier. In addition, bones can have similar (or higher) intensity values than contrast-enhanced blood vessels, and so it is necessary to segment out the bones before applying MIP. A variation on MIP is Closest Vessel Projection [286, 216], or Local MIP [223], where along each ray the first local maximum above a user-defined threshold is used.

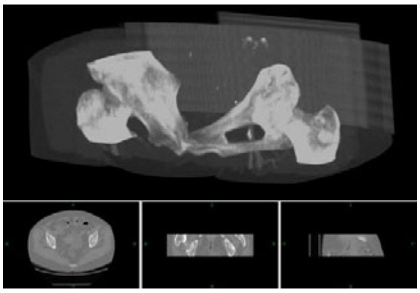


FIG. 3.28: *MIP volume rendering using VolView [132].*

3.3.4 Ray Tracing

This technique is widely documented in the literature and comprehensive reviews can be found in the following books [67, 273, 253]. A ray of light is traced from the observer (often called the camera) through each pixel in the image plane. In this context, "tracing" means that the intersections of the current ray with every object of the scene need to be computed in order to simulate different physical interactions between the ray and objects, and to compute the final colour of the pixel. In its basic implementation (*i.e.* without transparency), only the intersection with the object closest to the observer is useful; indeed other objects are not visible by the observer. Figure 3.29 shows an example of rendered image using the Persistence of Vision Raytracer (POV-Ray) implementation [197].

FIG. 3.29: *Example of image generated by ray-tracing using the Persistence of Vision Raytracer.*

Originally developed for image synthesis in the visible domain, it has been adapted for x-ray simulation (see Section 3.3.7).

3.3.5 GPU Programming

During the last several years, the usage of GPUs has widened within the computer industry. Graphics processors are based on a highly parallel architecture. Indeed, they are Single Instruction Multiple Data (SIMD) type processors [63, 213]. There is a growing interest for GPGPU and this has been an active area of research for years [147, 127, 187, 85]. Pioneering work takes advantage of the hardware acceleration using non-programmable GPUs, for example to compute

Voronoi diagrams [109] or wavelet transforms [101]. With the introduction of the programmability of GPUs, graphics hardware is considered as a "low-cost" parallel computing solution [272]. Using vertex and fragment programs written in a shading language – such as the C for Graphics (Cg) language by Nvidia, DirectX High Level Shader Language (HLSL) by Microsoft, or the OpenGL shading language (GLSL or GLSLang) by the OpenGL Architecture Review Board – it is possible to implement algorithms that run directly on the graphics vertex and fragment processors. Many research activities in computer graphics are taking advantage of the highly parallel architecture and the programmability of modern GPUs has raised, such as ray-tracing [32], radiosity [31], image processing [213], volume rendering [61], and many more [187]. In parallel, this also enables physicists to anticipate the realistic simulation of physics phenomenas [91]. Early work includes the simulation of phenomenas described using discrete partial difference equations (PDEs) or approximations of PDEs. The simulation of cellular automata is one example [115]. The simulation of natural phenomenas were also reported, such as the simulation of fluid or gas [92, 63, 282]. The first generation of programmable graphics hardware lacks floating point precision in the fragment processor. The pixel depth has now increased from 32 bits per pixel to 128 bits per pixel. Previously, each red, green, blue, and alpha component of a pixel were encoded in 8 bits. The intensity of each component was clamped between 0.0 and 1.0. In the graphics hardware, this range corresponds internally to $[0-255]$. Using 128 bits, each component of a pixel can now have 32-bit floating point precision. Negative values and values greater than "1" can thus be written into the frame buffer.

OpenGL ® Shading Language (GLSL) is a high-level programming language dedicated to the implementation of specific algorithms, shader programs, which will be executed directly on the GPU to replace the fixed graphics rendering pipeline. Figure 3.30 shows the OpenGL rendering pipeline (note that a fully detailed representation of the OpenGL rendering pipeline showing programmable processors for vertex and fragment shaders is available in the "OpenGL Orange Book" [213]). A shader program is twofold i) a vertex shader or (vertex program) substituting major parts of the vertex operations of the traditional fixed function of the geometry processing unit, and ii) a fragment shader (or fragment program) substituting major parts of the fragment operations of the traditional fixed function of the rasterization unit. Although commonly called "vertex shader" and "fragment shader", in this thesis we will prefer the terms "vertex program" and "fragment program" because the algorithm implemented here is not related to any "shading".

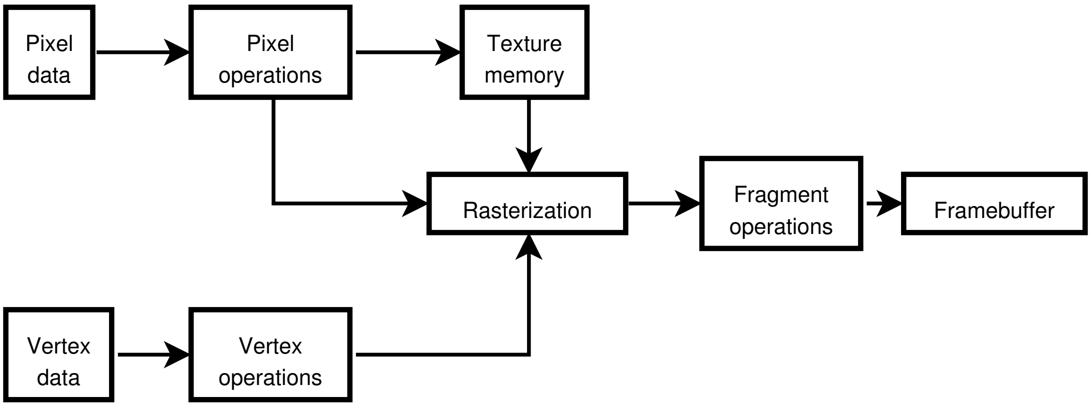


FIG. 3.30: *OpenGL rendering pipeline.*

3.3.6 Haptics

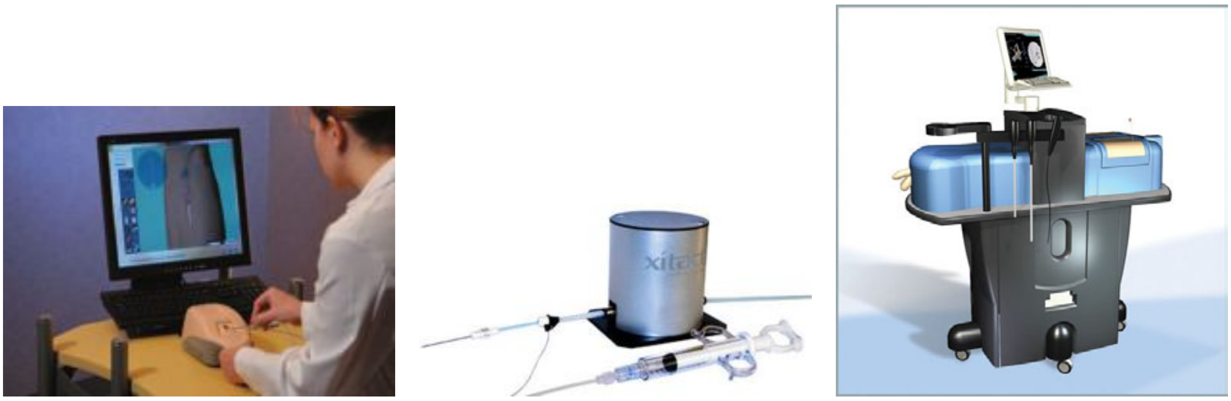
Haptics relates to the sense of touch. It can be divided into two main categories of sensory information. The first category relates to tactile information. It includes temperature, skin curvature and stretch, velocity, vibration, slip, pressure and local force. The second category relates to kinaesthetic information. Kinaesthetic rendering is often called force-feedback or force-reflection. Here a haptic interface is a force reflecting device that allows a user to touch, feel, manipulate, create and/or alter simulated 3D objects in a VE. Haptic hardware (see below) has only recently become affordable and are deployed in many areas, such as training VEs for medicine [247, 14]. The research presented in this thesis will address haptic rendering of ultrasound probes and needle puncture.

Devices

Laycock and Day proposed a useful review of haptic devices [142]. Here, we will focus on devices that are relevant to the simulation of IR procedures. Haptic devices that only allow the tracking of the position of the device in 3D provide 3 degrees of freedom (DOF) in position. Haptic devices that allow the tracking of both the position and rotation provide 6 DOFs in position. Similarly, devices that only apply a force on the position of the device provide 3 DOFs in force feedback and devices that also apply a force on the rotation of the device provide 6 DOFs in force feedback. We can distinguish two kinds of haptic devices. On one hand, there are devices designed for specific tasks, for examples in interventional radiology there are:

- *CathSim AccuTouch System* ® by *Immersion Medical* [110], a predecessor of
- the *Virtual I.V.* ™ system (see Figure 3.31(a)) by *Laerdal Medical* [140],
- *Xitact CHP* ™ (see Figure 3.31(b)) and *Xitact VSP* ™ by *Xitact* (now part of *Mentice Corporation*) [280],
- *Procedicus VIST* ™ by *Mentice Corporation* [166],
- *PERC Mentor* ™ (see Figure 3.31(c)) by *Simbionic* [242], and
- *ANGIO Mentor* ™ by *Simbionic* [242].

These provide haptic feedback for needles, catheters or guidewires (note that we can also add devices devoted to laparoscopic procedures to this list of specific haptic interfaces, see [225] for a brief overview of the currently available commercial products). The main interest of such dedicated devices is to accurately reproduce devices used in actual procedures. Although the *Virtual I.V.* ™ system is designed for use in training needle insertion, this device cannot be used into our simulator because it does not provide 6 degrees of freedom (DOF) in position that is required in the simulation of freehand techniques. Indeed, as illustrated in Figure 3.31(a), needles can only be inserted at a fixed point and their rotation is highly limited. This is the predominant drawback that prevents their use in our simulator because the trainee must be able to freely position and orientate the needle as he/she wants.



(a) *Virtual I.V.* by *Laerdal Medical*. *Source*: Reproduced from <http://www.laerdal.com/binaries/AFXLUAGZ.pdf>.

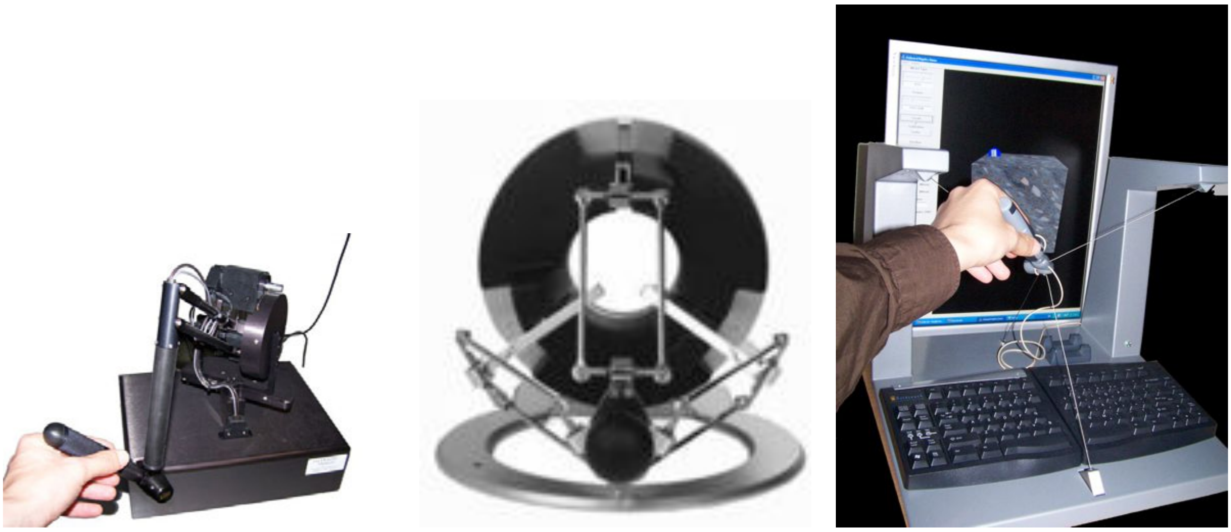
(b) *Xitact CHP* by *Xitact*. *Source*: Reproduced from <http://www.xitact.com/products/int/>

(c) *PERC Mentor* by *Simbionix*. *Source*: Reproduced from <http://www.simbionix.com/Brochures/PERC_Mentor.pdf>

FIG. 3.31: *Examples of commercial specific haptic devices.*

On the other hand, there are more generic devices that can be used, such as:

- the popular *PHANToM* ® family of devices (see Figures 3.32(a) and 6.15) developed by *SensAble Technologies* [233, 159, 217],
- the *Omega.x* family (see Figure 3.32(b)) by *Force Dimension* [68],
- the *Delta* family devices by *Force Dimension*,
- the *Falcon* by *Novint Technologies* [179], and
- the Mantis Workstation by Mimic Technologies (see Figure 3.32(c)) [170].



(a) *PHANToM 1.5/6DOF* by *SensAble Technologies* [235].

(b) *Omega 3* by *Force Dimension*. *Source*: Reproduced from <`http://www.forcedimension.com/fd/avs/home/products/omega_3/`>.

(c) *Mantis Workstation* by *Mimic Technologies*.

FIG. 3.32: *Examples of commercial generic haptic devices.*

They all provide 6 DOF in position: they can be freely moved and orientated in any position and orientation. The lowest priced devices can only provide 3 DOF in force feedback, *i.e.* the force feedback only constrains the position of the device, there is no force feedback constraining the torque of the device. They are suitable for the haptic rendering of single point probes, such as an ultrasound transducer. The most expensive devices provide 6 DOF in force feedback, *i.e.* the force feedback can constrain the position and the rotation of the device. They are well adapted to simulate the needle. Indeed, in the real world, once a needle is inserted deeply enough into the patient's soft tissues, it is not possible to rotate the needle around its tip, but its shaft can bend. However, the high cost of such devices still prohibit their use in an affordable medical simulator for training purposes. Section 5.2 describes the PHANToM Omni haptic device, by SensAble Technologies, which has been selected in this project to mimic both the needle and the US probe.

Force Rendering Algorithms

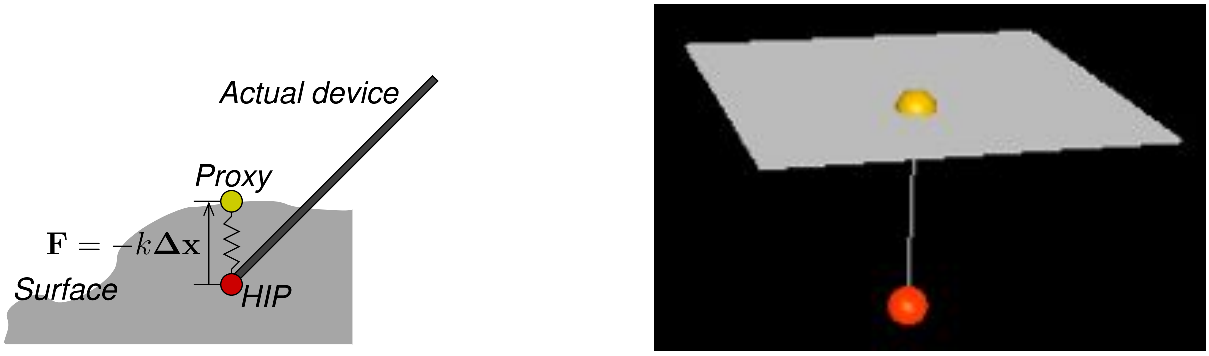
An early survey of haptic rendering methods is proposed by Srinivasan and Basdogan [247]. More recently, Laycock and Day presented a comprehensive survey [143]. To achieve acceptable tactile and force feedback perception during the simulation, it is widely accepted that the minimal haptic refresh rate must be 1 kHz [247, 220, 143]. The basic principle of haptic rendering is that when a collision between the haptic interface point (HIP) and any virtual 3D objects occurs, a force feedback will be computed and applied on the HIP. Indeed, when a collision between the HIP and an object occurs, the HIP is penetrating that object and the haptic rendering, in its basic form, will constrain the HIP to remain outside the object.

A proxy-based algorithm [215] is commonly accepted for the haptic rendering of surfaces. The haptic device interacts with the surface of a 3D object, usually represented as a triangular polygon mesh. The creation of suitable anatomy models from patient specific CT data is discussed in Section 4.2. In the proxy-based algorithm, a small sphere, the virtual proxy, represents the desired position of the device. Although the device can penetrate into the object with which it is interacting, the proxy always remains outside of its surface (see Figure 3.33). At each iteration of the algorithm, the proxy is moved in order to minimise its distance to the actual device. In other words the proxy is the point on the surface that is the closest to the actual HIP. Then, as shown by Figure 3.33(a), the surface will constrain the haptic device to reach the proxy using a force proportional to the distance (Δx) between the proxy and the actual device, the force feedback is then similar to a spring effect using the Hooke's Law:

$$\mathbf{F} = -k\Delta\mathbf{x} \tag{3.9}$$

Eq. 3.9 is often referred as simple spring equation. **F** is the response force. By manipulating the spring constant k, the stiffness of the object can be modified.

To mimic a stiff contact, the spring constant needs to be relatively large. However, haptic devices have a limited stiffness and often the spring constant cannot be sufficiently increased to perceive this effect and this may lead to instability. Salisbury *et al.* tackled this issue using the spring-



(a) When the haptic device penetrates the 3D object, the proxy remains on the surface and a spring force is returned to the user.

(b) Example of interaction between the PHANToM Omni and a plane. The proxy position is represented as a yellow sphere and the HIP as a red sphere. The force feedback is proportional to the distance between the spheres.

FIG. 3.33: *User interactions using the proxy-based algorithm.*

damper equation as follows [219]:

$$\mathbf{F} = -k\Delta\mathbf{x} + b\mathbf{v} \tag{3.10}$$

with $\mathbf{F}$ the response force, k the spring constant, $\Delta\mathbf{x}$ the displacement, b the damping coefficient, and $\mathbf{v}$ HIP's velosity. This helps to stabilize the force feedback.

Another common issue in haptic rendering is force discontinuities between successive iterations of the haptic rendering algorithm. This happens, for example, when the normal orientation rapidly changes during the rendering of a polygon mesh. Strategies analogous to Gouraud [84] and Phong shading [196] have been been adapted to interpolate normals during the haptic rendering [172, 219, 215]. In [143], a smoothing method that prevents large discontinuities is presented.

More recently, haptic rendering using voxel data has emerged. McNeely *et al.* proposed the *Voxmap-PointShell* method, a 6-DOF haptic rendering algorithm using volume data [164]. It was initially used to "preassemble" the Boeing 777. Objects of the scene are discretised into a grid of voxels. This is the *Voxmap*. Giving a state to each voxel of the Voxmap (free space, interior, on the surface, or within close proximity to the surface). Sets of points that lie exactly on the surface of the objects are labels, these are called *PointShell*. Using these, they integrate the collision detection, the force model, and the dynamic model into the haptic loop. Renz *et al.* proposed a method that ensures smooth surface representation to stabilize the haptic rendering of the Voxmap-PointShell method [208]. Petersik *et al.* adapted the proxy-based algorithm to use voxel data [194]. More recently, Palmerius *et al.* demonstrated a method that makes use of dynamic volume data to palpate a beating human heart [188].

Haptics in IR

The use of haptics is an important component in providing realistic manipulation of the virtual tools being used. Typically, the haptic model implemented is iteratively refined until it reaches a point when a domain expert is satisfied that the kinaesthetic sensation is good enough to mimic reality. Look Up Tables (LUTs) containing a sequence of empirically defined values is one approach that has been applied to needle insertion [79]. Each entry in the LUT contains the force value that is applied once the needle has reached a particular depth. Physically-based models that take into account the elastostatic properties of tissues under distortion have also been reported [48]. Recent work has focused on the measurement of forces during procedural instrumentation of real patients [49, 93]. The data can be used to create and validate a haptic model. In the Hermes project [204] a custom haptic interface for angioplasty has been built using such force measurement data in its design. The user experiences both forces and torque at their fingertips whilst manipulating a catheter like device. Mentice/Xitact also provide commercial haptic interface products that allow forces to be applied to catheters and guidewires (see Figure 3.31(b) for an example).

3.3.7 X-ray Simulation

The simulation of X-ray imaging techniques such as radiography or tomography is being extensively studied in the physics community with applications in crystallography, astronomy, non-destructive testing (NDT) and elsewhere. Different physically-based simulation codes are now available. Such programs are very useful in particle physics research. For example, by simulating experiments using realistic parameters, they allow scientists to optimize experiment parameters, to conceive new imaging systems, or to take into account non-destructive testing during the design of a mechanical structure [20, 148].

To date, there are two different kinds of X-ray simulation algorithms:

- probabilistic methods, based on Monte Carlo trials;
- determinist or analytic methods, based on ray-tracing.

Simulation by the Monte Carlo approach [20, 2, 222] consists of individually tracking each photon. Such algorithms are able to follow a photon during its different interactions with matter. At each step of the simulation, several events are possible for each photon. Events are chosen randomly with respect to physical law probabilities. The quality of such an algorithm depends on the quality of the random numbers generator. Monte Carlo simulation can produce very accurate X-ray images, but they are computationally intensive, which prevent their use in any interactive applications. For example, to simulate an image consisting of 10^6 pixels, with a noise level of 1%, at least 10^{10} photons have to be cast (depending on the attenuation in the object). This typically takes days of computation on a single PC.

To compute direct images (i.e. images formed by the X-ray beam transmitted without interaction through the scanned object), determinist methods based on ray-tracing techniques (see Section 3.3.4 for details on ray-tracing) provide a fast alternative to Monte Carlo methods [111, 144, 54, 8, 72, 73]. Here, all intersections between a ray and an object have to be considered and radiation attenuation is computed by considering the thickness penetrated by a ray into the object characterized by its material properties. The 3D scene is typically described by triangle meshes. This standard representation allows the use of computer-aided design (CAD) models (see Figure 3.34). Another reason to use triangle meshes is to make the render process faster. Indeed, many algorithms in real-time 3D computer graphics have been developed for such geometry representation, including polygon clipping and filling, etc. and also the classic Z-buffer algorithm to remove hidden faces [34]. A modified version of the Z-buffer, known as the *L*-buffer (for length buffer), can be used to store the length of a ray crossing a given 3D object [73].

The simulation of radiographic images from CT data sets has been also reported [111, 144]. More recently, volume rendering by ray-casting has been adapted to the realistic simulation of X-ray in a CAVE environment [151] and also to implement a hybrid determinist/probabilistic approach to compute the dose deposited in cancerous and healthy tissues during radiotherapy treatment [74]. In this case, each voxel corresponds to a cube characterized by its linear attenuation coefficient. From the X-ray source position, a ray is sent through each pixel of the simulated image. Using this approach, the attenuation of the incident X-ray beam is computed for each voxel traversed. Laney *et al* proposed a GPU simulation of X-rays based on volume rendering of unstructured data [141].

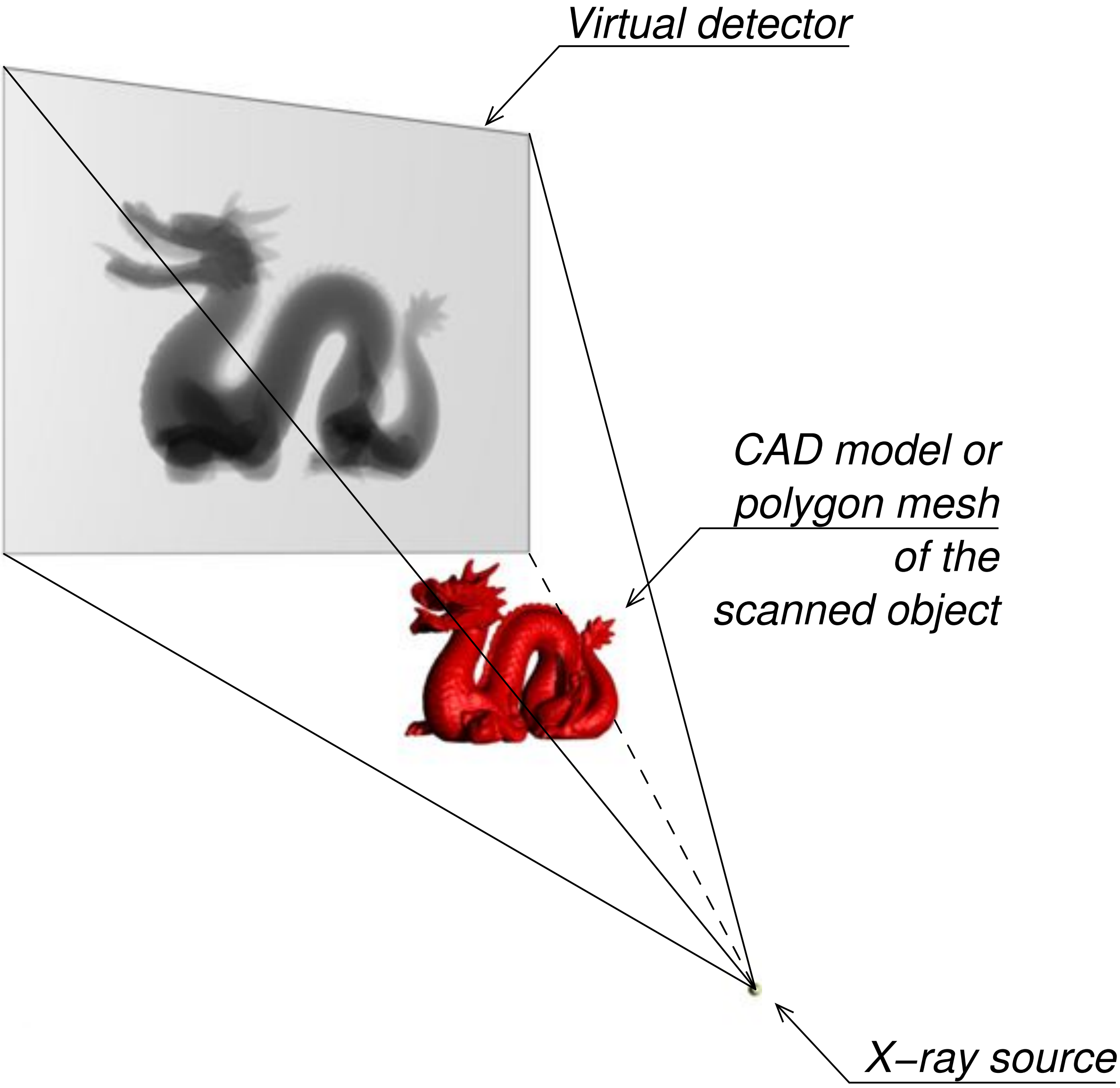


FIG. 3.34: *Radiographic simulation using a point source, a matrix detector and a CAD model with triangular meshes.*

3.3.8 Ultrasound Simulation

The simulation of ultrasound fields is also extensively studied in the physics community with applications in NDT and materials evaluation, in biomedical tissue characterization, in aided-design of ultrasound imaging systems, in studying transducers, in geophysics and elsewhere. The main approach relies on the concept of spatial impulse responses proposed by Tupholme and Stepanishen [262, 250, 251]. It is based on linear systems theory to calculate the ultrasound field for both the pulsed and continuous wave case [98]. The spatial impulse response of the transducer gives the emitted ultrasound field at a specific point in space as function of time. The convolution of the spatial impulse response by a given excitation function provides the ultrasound field for this excitation. The spatial impulse response varies depending on the position relative to the transducer. Jensen and Svendsen proposed a method to estimate this response for any transducer geometry and any apodization [118]. They divide the surface of the transducer into squares. The sum of the response of these squares provides the spatial impulse response of the transducer. They implemented this into *FIELD II*, a Matlab [260] toolbox to simulate the field of any ultrasound transducer [118, 117, 258]. Figure 3.35 is an ultrasound image of a kidney simulated using Field II. According to Ellis and Walker, DELFI's developers, the FIELD II software package is currently the standard for ultrasound simulation tools [59]. The *DELFI ultrasound simulation tool* is another Matlab toolbox used to simulation the field resulting from a given ultrasound system [59, 58, 57]. Contrary to FIELD II, DELFI is designed to efficiently compute the two-way spatial impulse response of ultrasound transducers [58]. *CyberLogic* provides commercial packages that make use of the finite difference method to efficiently solve the wave equation to simulate the propagation of ultrasound [12, 43]. Ultrasound wave propagation can also be simulated using the finite-elements time-domain (FETD) [23, 22]. A list of software packages and utilities for ultrasound research is compiled by the IEEE Ultrasonics, Ferroelectrics, and Frequency Control Society [108].

The real-time physically-based simulation of medical ultrasound images still remains a challenge and such an approach is not suitable in a real-time virtual environment. Indeed, tissues correspond to small non-uniform structures. When an ultrasound wave hits such an irregular surface, every of these obstacles behaves as a secondary sound source and it emits a wave in many directions. Alternatively, methods have also been proposed to visually approximate such images and this is the approach that we will use in this research [267]. Zhu *et al* proposed a method using texture mapping, artificial noise, and radial blurring [284]. Aiger and Cohen-Or proposed a method that makes use of 3D ultrasound datasets to generate 2D US images [3].

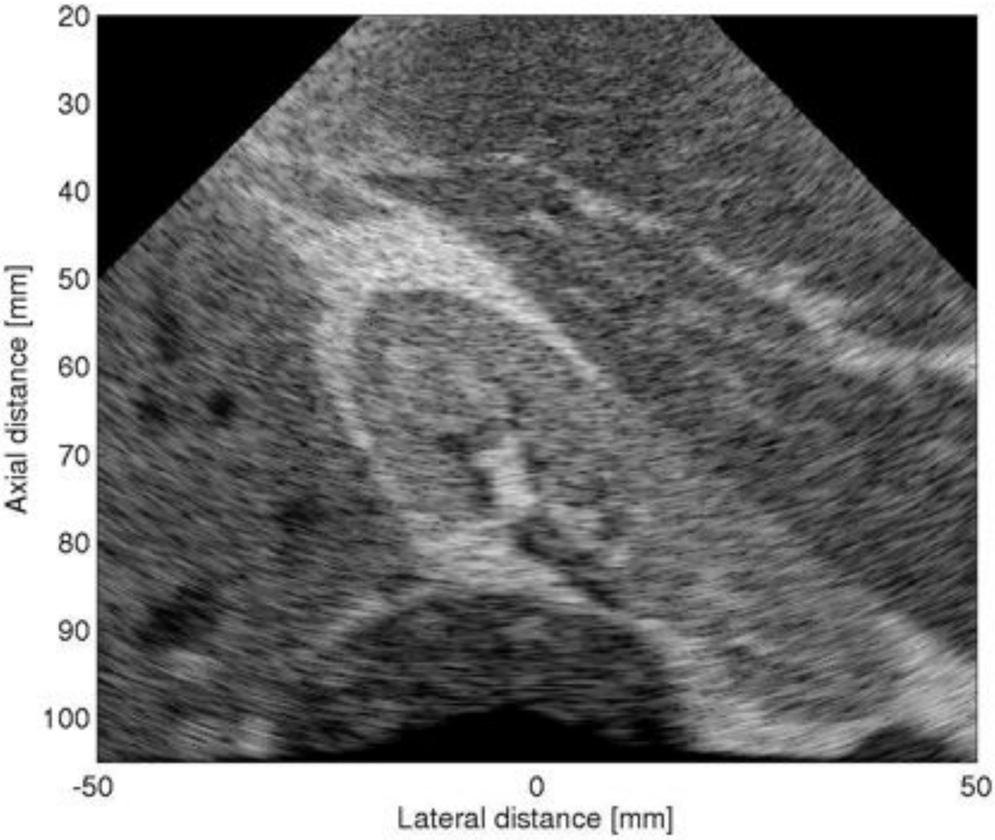


FIG. 3.35: *Calculation of B-mode image of synthetic kidney using Field II.* Source*: Reproduced from* `http://server.oersted.dtu.dk/personal/jaj/field/?examples/kidney_example/kidney_example.html`.

3.4 Applications in IR VEs

Computer based simulation can potentially overcome some of the limitations of fixed models (see Section 2.3); and computer-based simulations are predicted by many to be used soon more and more in surgery training [158]. The first generation of medical training simulators focused on laparoscopic surgery, which is well suited for computer simulation. Indeed, the requirement for completely new, specialist clinical skills, has driven the development of novel solutions to training [158]. Skills boxes and other bench models have been shown to accelerate the acquisition of skills in endoscopic procedures for urology [25, 24]. The development of a laparoscopic simulator model using VEs to train skills using, initially, simple geometric shapes, with subsequent development of more realistic visual displays and haptics [218, 246], showed for the first time that VEs could be a highly effective training tool [257]. Comprehensive surveys about VEs in medicine can be found in [161, 264]. The review below will focus on the medical field of interventional radiology.

More recently products have been introduced aimed at the IR domain such as the Mentice Procedicus VIST$^{\text{TM}}$ system for carotid and renal stenting; the Simbionix Mentor$^{\text{TM}}$ range including training for percutaneous access procedures with real-time fluoroscopy (*i.e.* x-ray); and Immersion Medical's AccuTouch Endovascular simulator, which includes support for percutaneous transluminal coronary angioplasty and stent. Often these commercial products have resulted from pioneering work carried out in academia; VIST for example has used the Interventional Cardiology Training System (ICTS) developed by the CIMIT Simulation Group [45]. These systems do not simulate the whole procedure, and few support needle puncture. They also tend to use predefined patient models, and do not yet provide the same fidelity of experience as training on a real person. Successful validation of IR simulators, for training and assessment in IR curricula, remains to be achieved [83]. In Figure 3.36, the trainees is practising guidewire insertion within a virtual environment.

The key role of laparoscopic surgery simulators, and also interventional radiology simulators, is in learning core skills, on which complex skills can be built. Such simulators have various advantages that are also common for both laparoscopic surgery and interventional radiology:

- they are comparatively low-cost;
- databases of cases can be implemented;
- the same exercise can be repeated many times by many students;

FIG. 3.36: *IR education with simulators.*

- the same student can rehearse a range of case mix that he/she would not normally encounter in a given time;
- exercises can be recorded, commented and analysed using predefined metrics; thought metrics evaluation by simulators is still at a rudimentary level. Indeed, key lower level metrics are only now being identified for IR [123];
- events such as drop of blood pressure, drop of heart beat frequency, breath, *etc.* can be triggered by the mentor or could be activated automatically at critical points in the procedure;
- they can help with skills maintenance of experts and for learning new and complex procedures, although they have not yet been validated for this;
- they can be use to rehearse beforehand a given procedure using patient specific data, *etc.* and maybe, for validation *vis-à-vis* insurance.

The idea that computer-based simulators combining virtual 3D anatomy models and haptics could also provide an alternative for training the core skill of image guided needle insertion is at the origin of this doctoral study. Indeed, this area has not been fully addressed by other medical VEs.

For image guided needle puncture, the Haptic Operative Realistic Ultrasonography Simulator (HORUS), developed at the IRCAD, is the only other computer based simulator to address this problem [102, 69], and is particularly aimed at ultrasound guided needle insertion for biopsy and radio frequency thermal ablation. As Figure 3.37 illustrates, two force feedback devices are used by HORUS: one for the ultrasound probe and one for the needle. The algorithm for simulating ultrasound images appears to be based on ray casting, but the implementation details have not been published. Zhu *et al* extend the use of a mannequin further (see Figure 3.38) [284]. They use a pair of calibrated magnetic 3D motion sensors to capture the position and orientation of a mock ultrasound probe and needle, whilst emulated ultrasound images are generated in real-time from a labelled volumetric data set that is non-rigidly aligned to the mannequin. Such an approach is relatively affordable (a pciBIRD card, by Ascension Technology Corporation [5], with one sensor costs between £1,700 and £4,200 depending on the chosen configuration []), though at the same time the learning process requires a complete range of anatomical and pathological encounters. There are difficulties in providing sufficient variety using solutions based on fixed anatomical models, and in addition the tactile properties and physiological responses of foam and other surrogate materials are far from that of a real person.

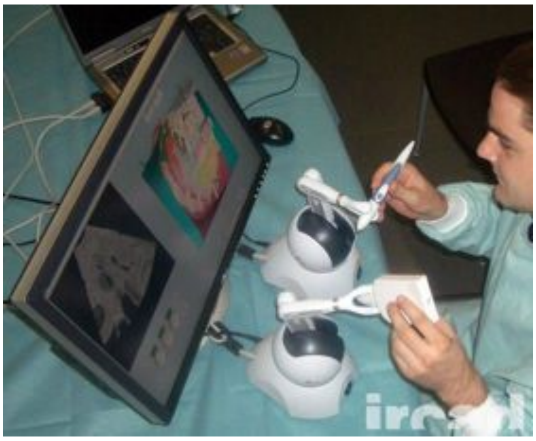


FIG. 3.37: *HORUS, the US guided needle puncture simulator developed IRCAD, France.* Source*: Reproduced from* `http://www.ircad.fr/virtual_reality/horus.php?lng=en`.

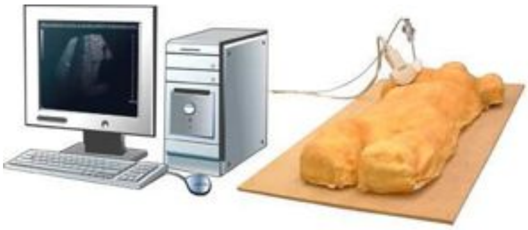


FIG. 3.38: *US guided needle puncture simulator developed at Leeds University, UK.* Source*: Reproduced from [284].*

3.5 Conclusion

Early simulators aimed at realism, but they were unsuccessful. Indeed, graphics hardware was too expensive, which prevented large-spread use, or was too slow, which did not make real-time simulations possible. One of the first successes probably was *MIST/VR* by Virtual Presence (see above). Its originality consisted of the task abstraction approach: manipulating simple 3D shapes using a laparoscopic user-interface.

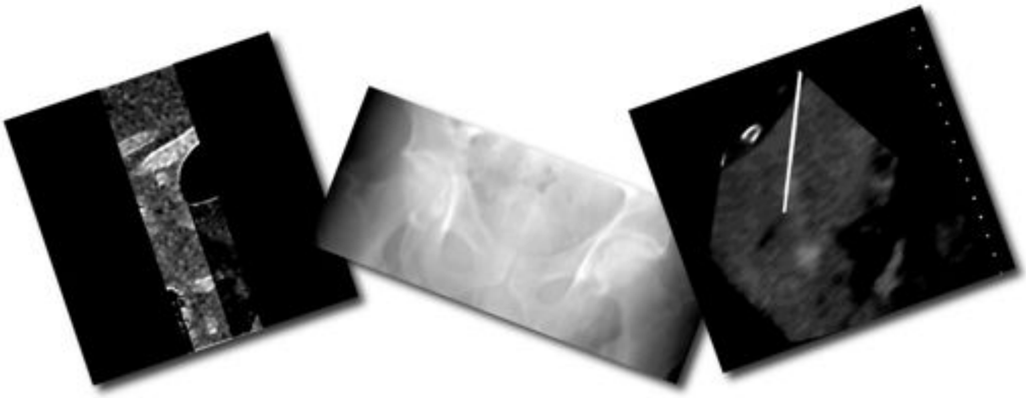
Nowadays, hardware price are becoming less of an issue and simulating realism is becoming possible although many research challenges still remain, such as realistic physical behavior of deformable tissue, fluid flow, haptic modelling, *etc.* Indeed, the sensation is a key issue in surgery skills learning or training by simulation. The future will also undoubtedly see more variety of solutions, not only confined to the field of laparoscopy surgery, such as interventional radiology, open surgery techniques, and other specialist domains such as dental surgery.

IR simulators start emerging on the market, but training of needle puncture guided by imaging (e.g. ultrasound, computed tomography or any other imaging modality) using the Seldinger technique - the first step in approximately half of all IR procedures - has been largely overlooked. The solution proposed in this thesis provides an alternative novel method of simulating ultrasound using the graphics processing unit (GPU). It also supports stereoscopic viewing, and different rendering schemes can be used to represent the real-time images displayed to the trainee. Our use of anonymised, real, patient specific imaging data also allows the training simulation to take place on different anatomical regions, using accurate and variable anatomical representation, and with a range of pathological entities. This variability is important to learning.

This thesis explores the possibilities given by off-the-shelf haptic devices and programmable graphics cards to respectively i) enable the trainee to feel the resistance of tissues on the needle, and the resistance of the skin and bones on the image probe, and ii) simulate both ultrasound images and fluoroscopic images from actual patient specific CT data, taking into account the balance between accuracy and computation time. This research has been carried out within a multi-disciplinary collaboration involving practising interventional radiologists, psychologists and computer scientists from the CR*a*IVE consortium [41].

Chapter 4

EFFICIENT VISUAL RENDERING USING HARDWARE ACCELERATION



Contents

4.1 Choosing the Input Data

Although one of the initial goals of this doctoral research was to deliver new techniques for achieving a cost-effective training tool for freehand ultrasound guided needle puncture, 3D ultrasound data sets are not used in our implementation. A 3D US data set is reconstructed from freehand 3D US images, the probe being tracked by electromagnetic sensors. Such 3D US data is not readily available in the hospital and x-ray CT undoubtedly remains the most commonly used modality for 3D data sets. However, this technique has been successfully tested [3]. In BIGN*e*PS*i*, the input data required is an anatomical CT scan specific to a particular patient's anatomy and pathology, although for ethical and Research Governance requirements, the identity of any patient is withheld and all imaging data is pseudo-anonymised. Ethical approval for this was obtained by the Royal Liverpool Hospital. Our use of pseudo-anonymised, real, patient specific imaging data also allows the training simulation to take place on different anatomical regions, using accurate and variable anatomical representation, and with a range of pathological entities. This variability is important to learning. Our clinical collaborators selected eight datasets of interest for training image guided needle puncture. These include kidneys, groin and liver region for visceral puncture and neck for central venous catheter (CVC). Then they are used to create the patient specific models (see Figure 4.1). This is also the data source for generating

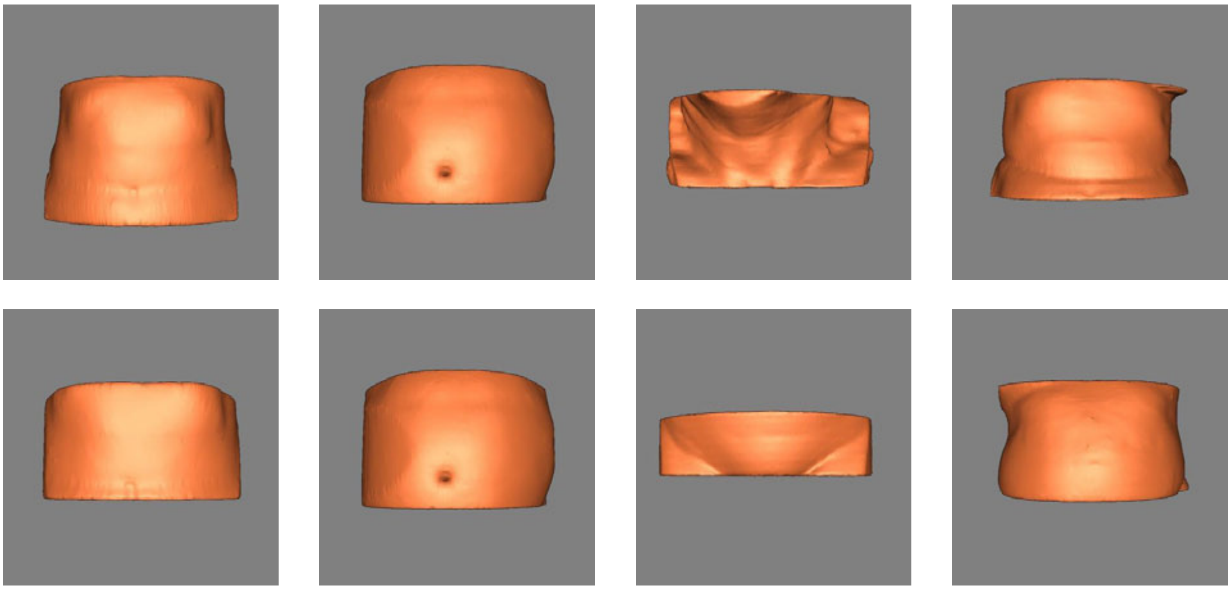


FIG. 4.1: *Isosurfaces of human skin generated from patient specific CT datasets selected by our clinician collaborators. Anatomical regions of interest include: kidneys, groin and liver region for visceral puncture and neck for CVC.*

real-time images that are used to guide the needle to the correct location. Different rendering schemes can be used to represent these real-time images. It includes i) MPR images of the CT volume, ii) ultrasound, and iii) fluoroscopy (see Sections 4.3, 4.6, and 4.7 respectively).

A smooth polygonal surface representing the relevant part of the patient's body must be generated to mimic the behaviour of an image probe moving over the skin of the patient. The extraction of the skin surface from 3D US is a challenging problem, however, due to the noise and artefacts in the data. Moreover, our volume haptic models for the image probe and needle insertion (see Chapter 5) rely on the voxel intensities when calculating resistance according to the tissue properties at the probe position or at the needle tip respectively. The range of intensities from 3D US is only $[0,255]$, this has proved unsuitable for this approach. Indeed, this is not possible to accurately distinguish the kind of tissues at the tip of the needle.

CT has proven to be a valid data source for our simulator. It is commonly used in hospitals for diagnosis, and provides reconstructed volume data of internal structures of the human body at a relatively high resolution compared to ultrasound data. The resulting stack of 2D trans-axial slices defines a voxel data set. Medical scanners deliver this data using the DICOM standard. Patient specific CT data is therefore theoretically easy to obtain, though first requires careful attention to research ethics and research governance requirements. For more details, an historical retrospective and the physical background of x-ray imaging, including CT, were given in Sections 3.1.1 and 3.2.1 respectively.

4.2 Input Data Pre-processing and Generating Patient Specific Models

A choice has been made to directly use the CT volume data as the input data of our simulator. Indeed, one requirement was to ease the preparation of new specific patient datasets for being used in our simulator. When possible, fully automatic processing of the data has been preferred to semi-automatic or manual processing. This avoids segmenting tissues and organs for each patient's dataset. Instead values in Hounsfield unit (HU) are directly used when applicable. Processing of the patient's specific image data has been implemented using ITK and VTK when applicable [281, 228]. The result of each time-critical and invariant pre-processing step is stored in files because the simulation using a given patient specific dataset can start to read the data from these recorded files faster.

4.2.1 Volume Data Pre-processing

In the case of a GPU-based implementation of the image rendering algorithms, to be accessible by the fragment program, the volume dataset must be stored directly into the memory of the graphics card as a 3D texture. For this purpose, the data has to be rescaled onto the $[0.0, 1.0]$ scale. The haptic models presented in Chapter 5 make use of the gradient amplitude of the patient dataset for collision detection. Computing the rescaled volume and the gradient amplitude volume are computationaly intensive and are invarient. Figure 4.2 describes the pipeline that has been implemented using the Insight Segmentation and Registration Toolkit (ITK) [281]:

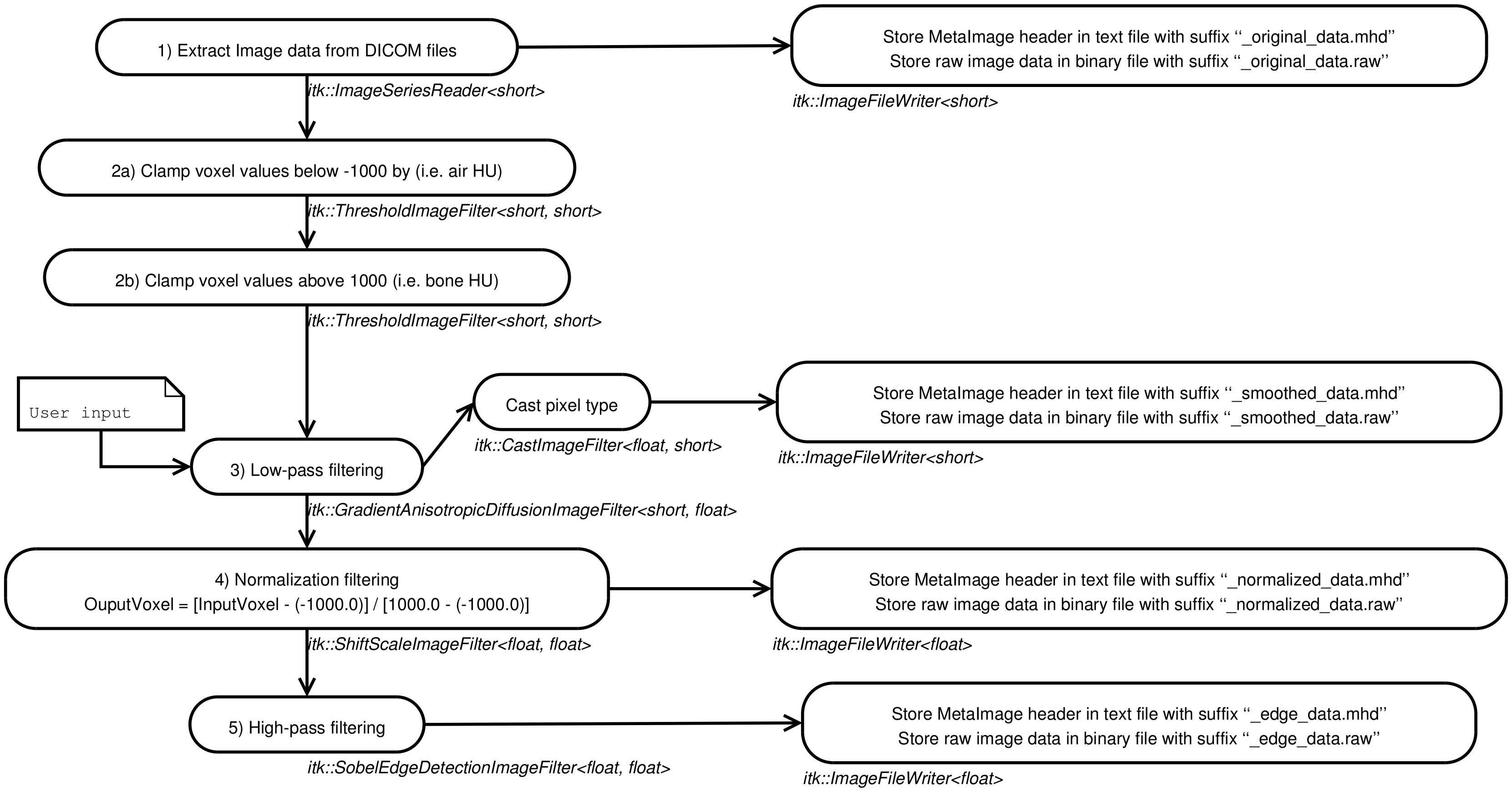

1) Extract Image data from DICOM files
itk::ImageSeriesReader<short>
Store MetaImage header in text file with suffix "_original_data.mhd"
Store raw image data in binary file with suffix "_original_data.raw"
itk::ImageFileWriter<short>
2a) Clamp voxel values below -1000 by (i.e. air HU)
itk::ThresholdImageFilter<short, short>
2b) Clamp voxel values above 1000 (i.e. bone HU)
itk::ThresholdImageFilter<short, short>
User input
3) Low-pass filtering
Cast pixel type
itk::CastImageFilter<float, short>
Store MetaImage header in text file with suffix "_smoothed_data.mhd"
Store raw image data in binary file with suffix "_smoothed_data.raw"
itk::ImageFileWriter<short>
itk::GradientAnisotropicDiffusionImageFilter<short, float>
4) Normalization filtering
OuputVoxel = [InputVoxel - (-1000.0)] / [1000.0 - (-1000.0)]
Store MetaImage header in text file with suffix "_normalized_data.mhd"
Store raw image data in binary file with suffix "_normalized_data.raw"
itk::ShiftScaleImageFilter<float, float>
itk::ImageFileWriter<float>
5) High-pass filtering
itk::SobelEdgeDetectionImageFilter<float, float>
Store MetaImage header in text file with suffix "_edge_data.mhd"
Store raw image data in binary file with suffix "_edge_data.raw"
itk::ImageFileWriter<float>

FIG. 4.2: *Pre-processing of the original X-ray CT datasets: 1) the input volume data is extracted from DICOM files, 2) the intensity of voxels is clamped within the range* $[-1000, 1000]$*, 3) a smoothing filter preserving edges is applied to reduce the amount of noise, 4) pixel values are linearly modified from the range* $[-1000, 1000]$ *to the range* $[0.0, 1.0]$*, 5) a high-pass filter is applied to detect edges.*

1) Extracting 3D data from DICOM files: loads patient specific data from native medical scanner files;

2a) Thresholding clamps voxel values below -1000;

2b) Thresholding clamps voxel values above 1000;

3) Low-pass filtering using an anisotropic diffusion algorithm reduces the level of noise in the data and also preserves edges [276];

4) Normalization filtering linearly rescales values from $[-1000, 1000]$ scale to $[0.0, 1.0]$ scale;

5) High-pass filtering detects interfaces between tissues of different kinds.

An exhaustive description of the pre-processing steps is available in Appendix B.1.

4.2.2 Generating Patient Specific Anatomy Models

Although the CT volume data is directly used in BIGN*e*PS*i* as the input data, a pre-processing step is applied to segment models of the patient's skin surface using the well known marching cubes algorithm [156]. This represents the surface as a triangular mesh and can be exported into a suitable file format, for example, using the standard Virtual Reality Modeling Language (VRML). Using polygon models in the virtual environment screen also allows the whole procedure to carried out in stereo and to efficiently compute shadows of the patient's body, of the image probe and of the the needle. This helps the trainee to improve positioning of the needle and probe at the skin entry site. The Marching cubes algorithm produces a high number of triangles. This is one of the main drawbacks of this algorithm. Therefore we summarize below a strategy to cut down the number of triangles in the polygon models (see Appendix B.3 for a complete description). Figure 4.3 describes the pipeline that has been implemented using the Visualization ToolKit (VTK) [228]:

Read the volume dataset

Thresholding clamps voxel values to the $[-1000, 1000]$ scale;

Resampling the volume reduces the volume resolution so that the Marching cubes algorithm will produce fewer polygons;

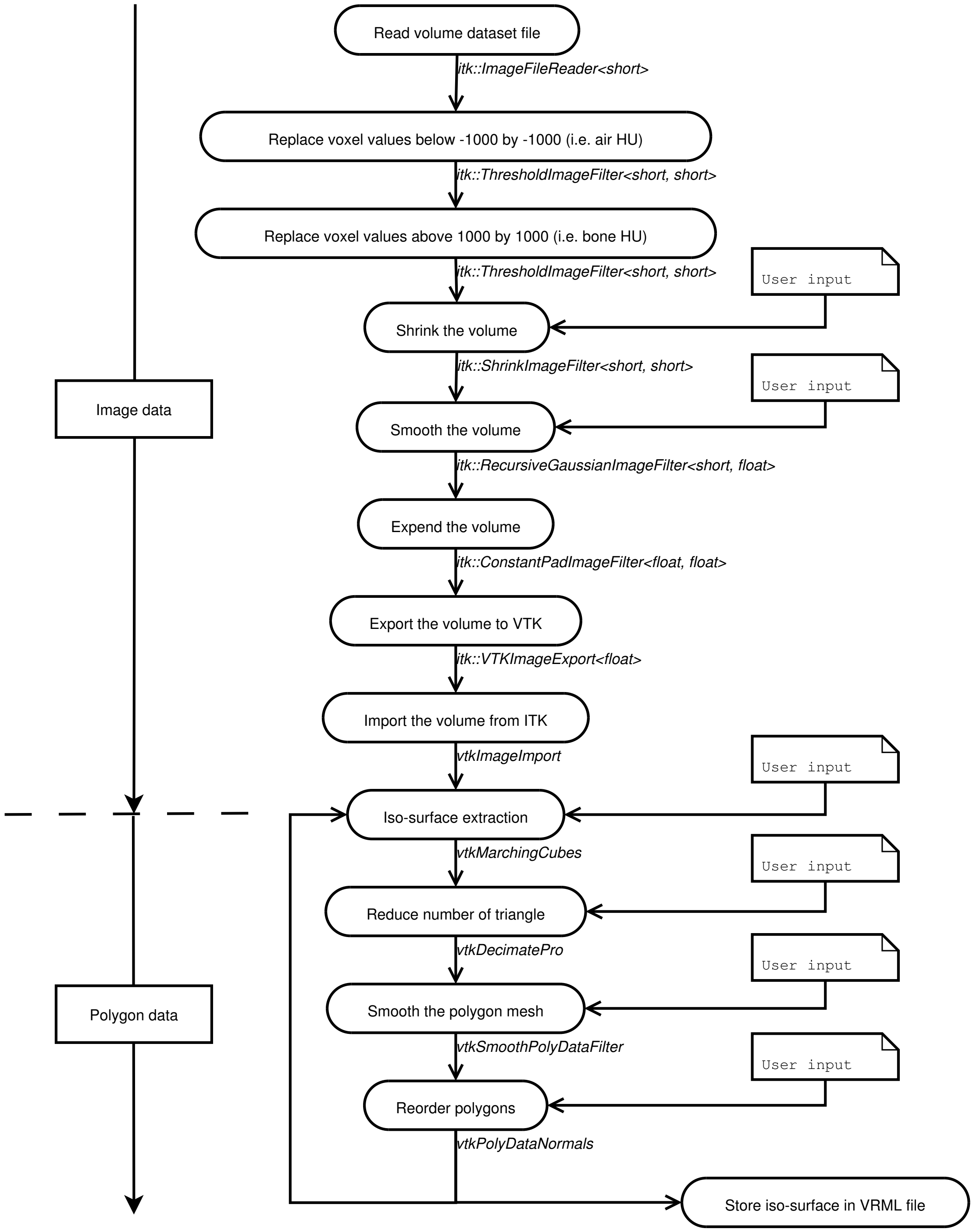

Read volume dataset file
itk::ImageFileReader<short>
Replace voxel values below -1000 by -1000 (i.e. air HU)
itk::ThresholdImageFilter<short, short>
Replace voxel values above 1000 by 1000 (i.e. bone HU)
itk::ThresholdImageFilter<short, short>
User input
Shrink the volume
itk::ShrinkImageFilter<short, short>
User input
Image data
Smooth the volume
itk::RecursiveGaussianImageFilter<short, float>
Expend the volume
itk::ConstantPadImageFilter<float, float>
Export the volume to VTK
itk::VTKImageExport<float>
Import the volume from ITK
vtkImageImport
User input
Iso-surface extraction
vtkMarchingCubes
User input
Reduce number of triangle
vtkDecimatePro
User input
Polygon data
Smooth the polygon mesh
vtkSmoothPolyDataFilter
User input
Reorder polygons
vtkPolyDataNormals
Store iso-surface in VRML file

FIG. 4.3: *Detailed description of the pipeline implemented to generate iso-surfaces from X-ray CT datasets.*

Expanding the data adds one voxel corresponding to air to each side of the dataset so that the Marching cubes algorithm will close the segmented surface to the external boundary of the volume dataset if needed;

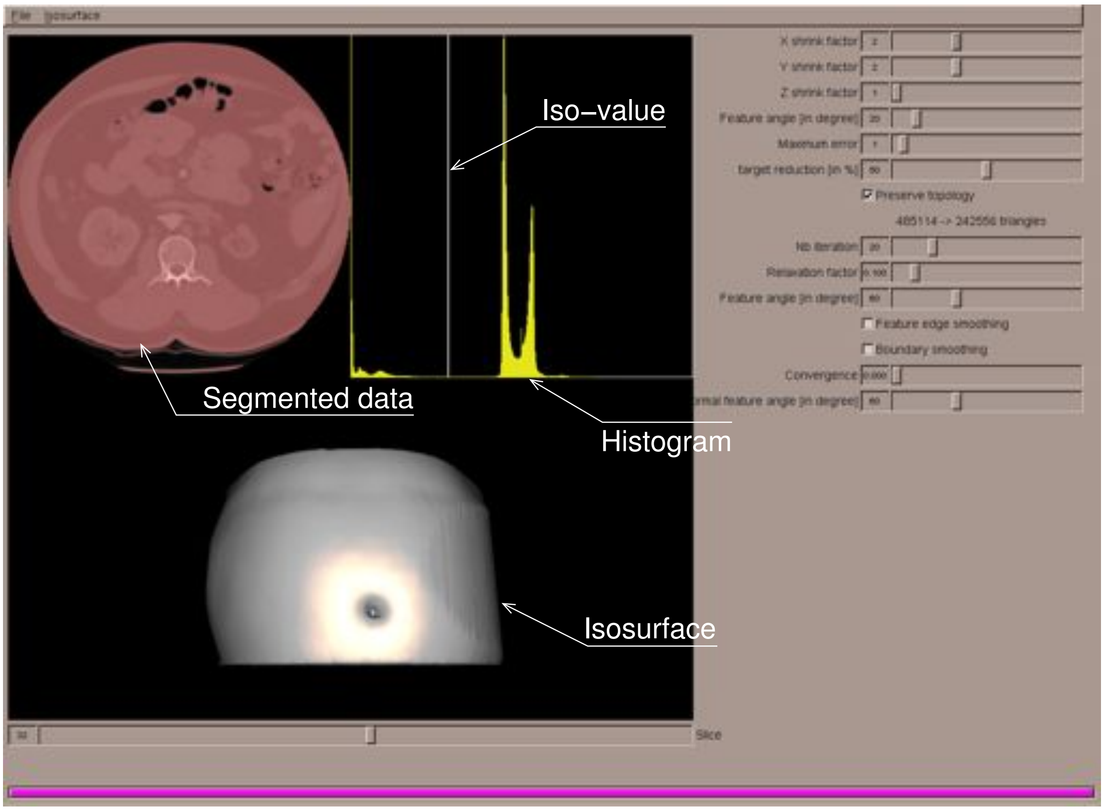
Iso-surface extraction uses the Marching cubes algorithm to generate a polygon model of an iso-surfuce;

Decimation filter reduces the number of polygons;

Smoothing the polygon mesh reduces, using Laplacian smoothing filter, the "wood-grain artefact" that is due to the low sampling rate of the volume dataset;

Polygon reordering consistently corrects the orientation of normal vectors.

Figure 4.4 shows the GUI of the implemented tool. It has been developed using ITK, VTK and FLTK [113, 133, 66]. On the right-hand side, the user can manipulate parameters of VTK filters. On the black area that lies on the left-hand side, the user can see the segmented image on



X shrink factor 2
Y shrink factor 2
Z shrink factor 1
Feature angle (in degree) 20
Maximum error 1
target reduction (in %) 50
☑ Preserve topology
485114 -> 242556 triangles
Nb iteration 20
Relaxation factor 0.100
Feature angle (in degree) 60
□ Feature edge smoothing
□ Boundary smoothing
Convergence 0.000
rmal feature angle (in degree) 60
32

FIG. 4.4: *Screen capture of the implemented tool to generate patient specific anatomy models.*

the top the original image. He/she can visualize the histogram of the volume dataset and select the Marching cubes iso-value by manipulating, using the mouse, a threshold on the top of the displayed histogram.

An alternative method would be to represent the virtual patient by volume rendering the CT data set. However, this technique proved to be too slow for use in our simulator, even when using a powerful graphics card. Volume rendering also still requires pre-segmentation of the internal organs of interest. Moreover, shadowing using hardware shadow maps can be implemented with polygon models, and our haptic models are hybrid models making use of volume data and taking advantages of polygon data (see Chapter 5).

4.3 2D MPR image of CT volumes

As the trainee moves the image probe over the skin of the virtual patient, an image corresponding to the position and orientation of the image plane is computed and displayed in realtime on the left-hand side of the virtual environment (see Figure 4.7). The goal is to display the image using a variety of formats, including CT, US and fluoroscopic. For the rendering of CT and US images, the first step is to use a 2D multiplanar reconstruction (MPR) image, *i.e.* as an oblique image computed from the original set of parallel CT slices, as illustrated by Figure 4.5. The MPR image can be directly computed by "reformatting" the input CT data, but without hardware support this technique has proved to be too slow for updating images in realtime with a

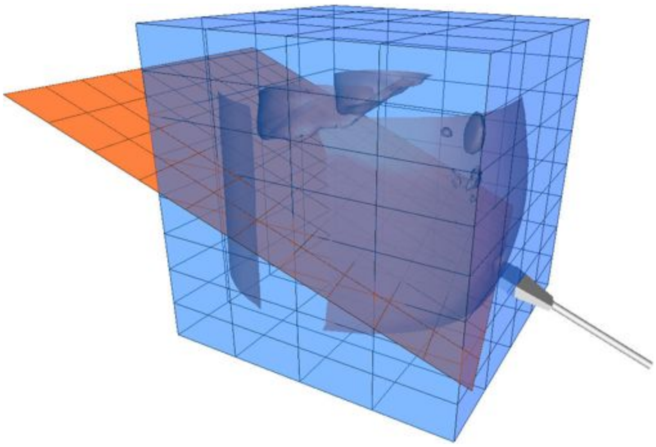


FIG. 4.5: *Oblique 2D MPR image generated from a set of parallel slices according to the orientation and position of the image probe – in blue the volume dataset – in orange the 2D MPR image.*

good resolution. Modern graphics cards support 3D texturing, which can be exploited to provide hardware acceleration for computing the MPR image. A flexible scheme for generating the images with full hardware acceleration has been implemented using OpenGL, a cross-platform standard 3D graphics application programming interface (API).

During the procedure the trainee uses the displayed image to guide the needle puncture, and so the needle must appear on the image as it is being inserted. This can be achieved by replacing the original voxel intensities of the CT dataset with a high value corresponding to the metal material of the needle shaft. Nevertheless, it is not possible to update the content of the 3D texture in realtime to change the voxel intensities at the needle position because of the overhead involved in transferring the recalculated data from main memory to the graphics card. The following section describes how to address this limitation using programmable graphics hardware.

4.4 Updating 3D Textures to track the Needle

4.4.1 Preliminary

To compute 2D MPR images, US images or fluoroscopic images, the needle must be taken into account when it is inserted into the patient's tissues and it must be displayed when required. The algorithms used to compute these images are presented in Sections 4.3, 4.6 and 4.7 respectively. These make use of a 3D texture to store the patient volume dataset into the graphics card memory, so that graphics hardware acceleration can be used to compute the images.

The naive approach to visually displaying the needle's position consists of checking every voxel of the dataset, for each computed frame, in order to:

(i) detect voxels which were not occupied by the needle during the previous iteration but are now, and to replace their original intensity by a high value corresponding to metal using Hounsfield units;

(ii) detect voxels which were occupied by the needle during the previous iteration and that are now outside of the needle, and to restore their original values. This is useful, for example, when the trainee is shaking the needle to locate its tip.

Using the OpenGL function **glTexSubImage3D**, it is possible to change the intensity of an arbitrary voxel of a 3D texture [182]. However, using this naive method, it is not possible to update the voxel intensities corresponding to the needle position in realtime because of the overhead involved in transferring the recalculated data to the graphics card.

An important new development in graphics programming, however, is the use of high-level programming languages to take control over critical stages of the graphics rendering pipeline, including 3D texturing (see Section 3.3.5). Indeed, using the new OpenGL® Shading Language, it is now possible to avoid the limitations of the fixed graphics rendering pipeline and to take advantage of programmable graphics hardware and of their parallel architecture. Then it becomes possible to detect if the texture coordinates in the 3D texture are either being penetrated by the needle or not. For each fragment that is being processed by the fragment processor, it is possible GLSL to perform computations on the graphics processing unit (GPU) to detect if the texture coordinates corresponding to a given voxel, computed from the 3D texture, are either being penetrated by the needle or not. Only the texels that do not belong to the needle will be retrieved. The following section gives details about the algorithm.

4.4.2 Principles

A needle can be considered as a finite cylinder. A 3D point is an interior point of an infinite cylinder if the shortest distance between this point and the cylinder axis is less than or equal to the radius of that cylinder. Using trigonometric identities it is possible to efficiently detect whether texture coordinates corresponding to the voxel V in the original CT dataset is an interior point of the infinite cylinder defined by its radius (*radius*), its position (T) and its direction vector (**DirNeedle**) (see Figure 4.6). Indeed, we know from the Pythagoras Theorem that:

$$|\mathbf{TV}| = \sqrt{|\mathbf{VW}|^2 + |\mathbf{TW}|^2} \quad (4.1)$$

$$|\mathbf{VW}|^2 = |\mathbf{TV}|^2 - |\mathbf{TW}|^2 \quad (4.2)$$

with W the orthogonal projection of T onto the infinite line defined by T and **DirNeedle**.

Let α be the angle between **DirNeedle** and **TV** (the vector from the needle's tip to the voxel of

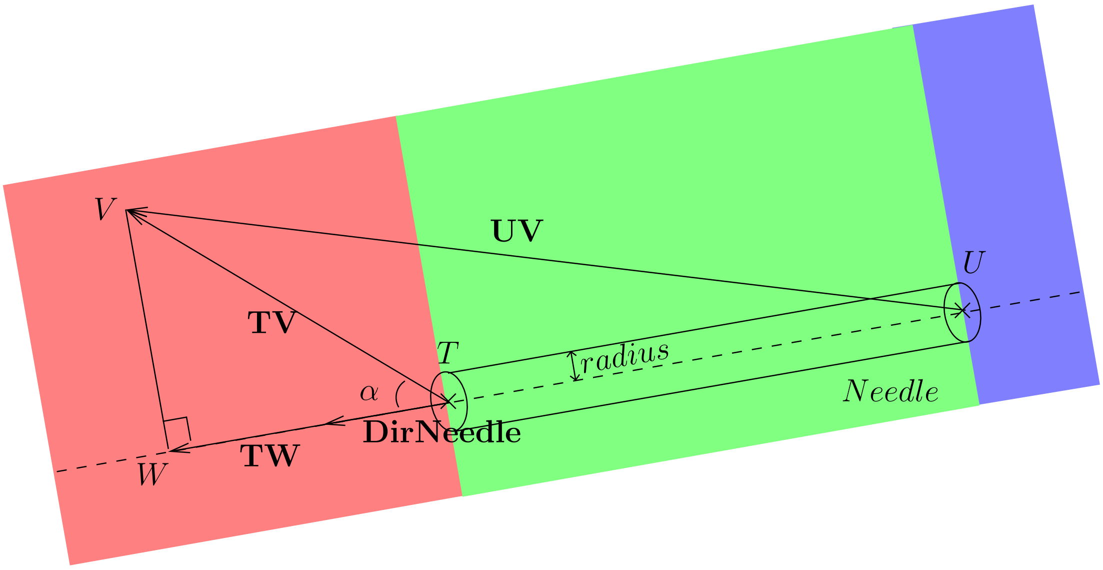


FIG. 4.6: *Needle/voxel intersection (with T as the position of the needle tip, U as the position of the extremity of the needle shaft,* **DirNeedle** *is the needle direction, and V is the voxel position).*

interest). It follows that:

$$\mathbf{TV}.\mathbf{DirNeedle} = \cos\alpha \times |\mathbf{TV}| \times |\mathbf{DirNeedle}| \tag{4.3}$$

$$\cos\alpha = \frac{\mathbf{TV}.\mathbf{DirNeedle}}{|\mathbf{TV}| \times |\mathbf{DirNeedle}|} \tag{4.4}$$

$$\cos\alpha = \frac{\mathbf{TV}.\mathbf{DirNeedle}}{|\mathbf{TV}|} \tag{4.5}$$

and also:

$$\cos\alpha = \frac{|\mathbf{TW}|}{|\mathbf{TV}|} \tag{4.6}$$

Thus we have:

$$|\mathbf{TW}| = \mathbf{TV}.\mathbf{DirNeedle} \tag{4.7}$$

We can finally deduce $|\mathbf{VW}|$ from Eqs 4.2 and 4.7:

$$|\mathbf{VW}|^2 = |\mathbf{TV}|^2 - (\mathbf{TV}.\mathbf{DirNeedle})^2 \tag{4.8}$$

In detail, a voxel should be marked as a needle voxel if it satisfies all of the three following conditions:

(i) if the distance |**VW**| is less than or equal to the needle radius then *V* is interior to the infinite cylinder defined by the needle position, direction and radius;

(ii) if the vector dot product between **TV** and **DirNeedle** is negative, then the voxel *V* is behind the tip of the needle, *i.e* it belongs to the green or blue areas in Figure 4.6;

(iii) similarly, if the vector dot product between **UV** and **DirNeedle** is positive, then the voxel *V* is at the front of the other extremity of the needle, *i.e.* it is located into the red or green areas in Figure 4.6.

4.4.3 Implementation

This algorithm has been implemented using GLSL. As it takes over the fixed graphics rendering pipeline, texturing operations must be implemented using a vertex program and a fragment program. In summary, in the fragment program, before fetching any value of the 3D texture, the texture coordinates are tested to check if the corresponding texel should be occupied by the needle. If so, there is no need to fetch the value from the texture and a high intensity is then used. On the other hand, if the corresponding voxel is not occupied by the needle, then the texel is fetched as normal. Appendices B.4.1 and B.4.2 are implementation examples of the vertex program and fragment program of the proposed method. Figure 4.7 illustrates a needle being inserted into a MPR image.

4.5 Updating 3D Textures to add a Lesion into the Patient

This section describes how to add some pathology for training image guided needle biopsy, *i.e.* a target corresponding to a lesion. It is then also possible to record metrics, *e.g.* if the target has been reached, the closeness of the needle tip to target periphery (the trainee can press the device's button of the needle once he/she thinks to have reached the target), the path of the needle, the number of needle withdrawals/reinsertions before entering the target, percentage of time that the target and needle are simultaneously visualised in the ultrasound beam, *etc.*

The content of the 3D texture needs be updated to add the lesion at the desired position. This needs to be performed before running the simulation. The volume dataset containing edge data, which is used for collision detection in our haptic model of needle insertion, also needs to be updated before the simulation starts. Alternatively, these two volume datasets can be efficiently

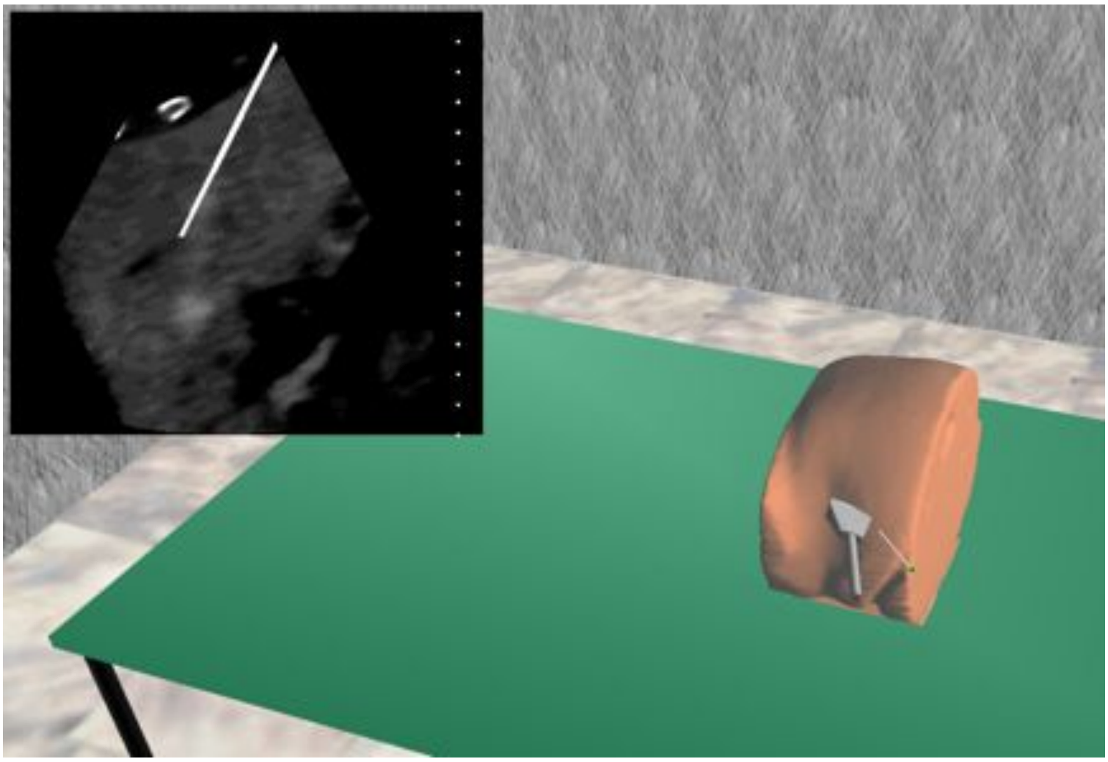


FIG. 4.7: *The position and orientation of the US probe in the VE defines a cutting plane that is displayed on the top left-hand side corner. This is a MPR image computed from a 3D texture that is updated in realtime, using GLSL for the computations, to insert the needle (white trace).*

"modified" at run-time level. In the previous section, we saw how to insert a finite cylinder into a 3D texture. A similar method can be used to insert a sphere that corresponds to a lesion into the patient. For each fragment that is being processed by the fragment processor, it is possible to perform computations on the graphics processing unit (GPU) to detect if the texture coordinates corresponding to a given voxel (V), computed from the 3D texture, are either located into a sphere corresponding to the lesion or not (see Figure 4.8). Only the texels that do not belong to the lesion will be fetched. In detail, a voxel should be marked as a lesion voxel if it satisfies this condition:

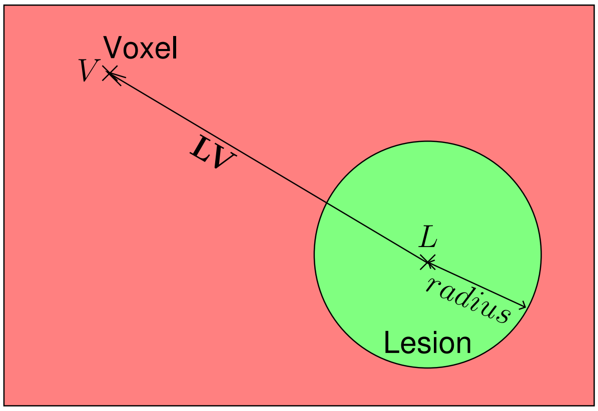


FIG. 4.8: *Lesion/voxel intersection (with L as the position of the centre of the lesion, and V as the voxel position).*

- if the distance $|\mathbf{LV}|$ between the voxel position V and the centre of the lesion L is less than or equal to the lesion radius then V is interior to the lesion defined by its centre and and its radius.

Appendix B.5 is an implementation example of the fragment program of the proposed method. Figure 4.9 is a US-like image that illustrates a lesion inserted into the 3D texture. Care must also be taken in the haptic rendering of the needle puncture to perform collision detection between the needle tip and the lesion (see Section 5.3.2).

4.6 Approximating Ultrasound Images

In Section 3.2.2, we saw some characteristics of ultrasound images. These include speckle noise, acoustic shadowing, specular reflections at the interface of a smooth large surface and enhancement. A flexible scheme for generating the US-like images with full hardware acceleration when

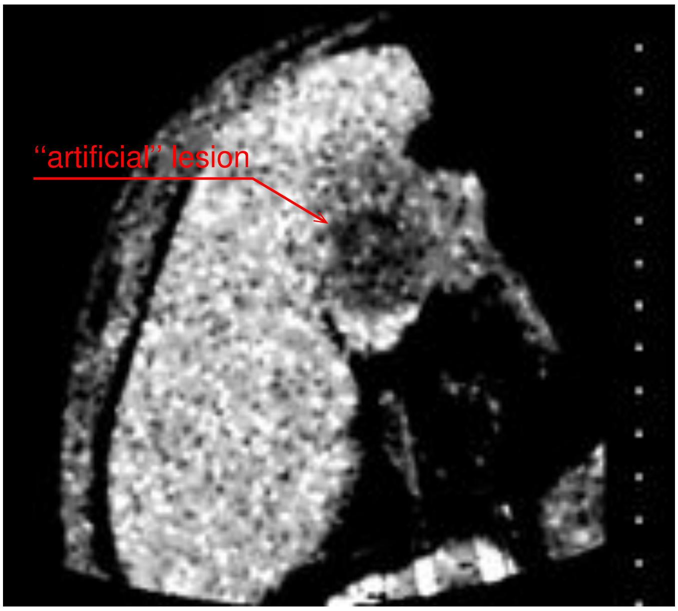


FIG. 4.9: *2D US-like image. The mentor has inserted a spherical “artificial” lesion into the patient anatomy. The lesion appears darker than surrounding tissues. This is the target for a needle biopsy training.*

applicable is explained below. Its rendering pipeline is illustrated in Figure 4.15. It is implemented using OpenGL.

4.6.1 2D Multiplanar Reconstruction Image

For US images, the first step is to use a 2D MPR image, *i.e.* an oblique image computed from the original set of parallel CT slices. Section 4.3 describes how this MPR image can be efficiently computed on the GPU using a 3D texture. During the procedure the operator/trainee uses the displayed image to guide the needle puncture, and so the needle must appear on the image as it is being inserted. This can be achieve efficiently using GLSL. Section 4.4 describes an effective method to achieve this by assigning to those voxels penetrated by the needle a high value corresponding to the metallic material of the needle shaft. Tissue lesion can also be added into the patient using a relatively similar method. In practice, the MPR image is in RGB mode. The red channel corresponds to the MPR data computed from the 3D texture. The green channel is a binary mask. The value of the green channel is “0” when the pixel is not into the needle, and it is “1” when it is in the needle. This allow us to differentiate between pixels that correspond to bones and those that correspond to the needle. This is used later to compute shadowing in ultrasound images.

In interventional radiology, linear probes or curvilinear probes can be used to guide a needle. When a linear probe is used, the 2D MPR image produced is a rectangle. Its width directly depends on the actual probe size. The width of the probe commonly used by our clinician collaborators is approximately 8 centimetres. The height of the image can be modulated by the user.

Figure 4.10 illustrates that the shape of the image is no longer a rectangle when the probe used to generate the ultrasound image is curvilinear. This feature needs to be integrated into the simulator. In Figure 4.10, the “field of view” (α) and distance l_1 are constant values that depend on the curvilinear probe in use. α and l_1 are estimated from a real ultrasound image that has been acquired using the curvilinear probe that our interventional radiologist collaborators use in the operation room. By contrast, the distance l_2 can be changed manually by the radiologist. In the simulator, this value can be interactively modified by the trainee. The shape of the ultrasound

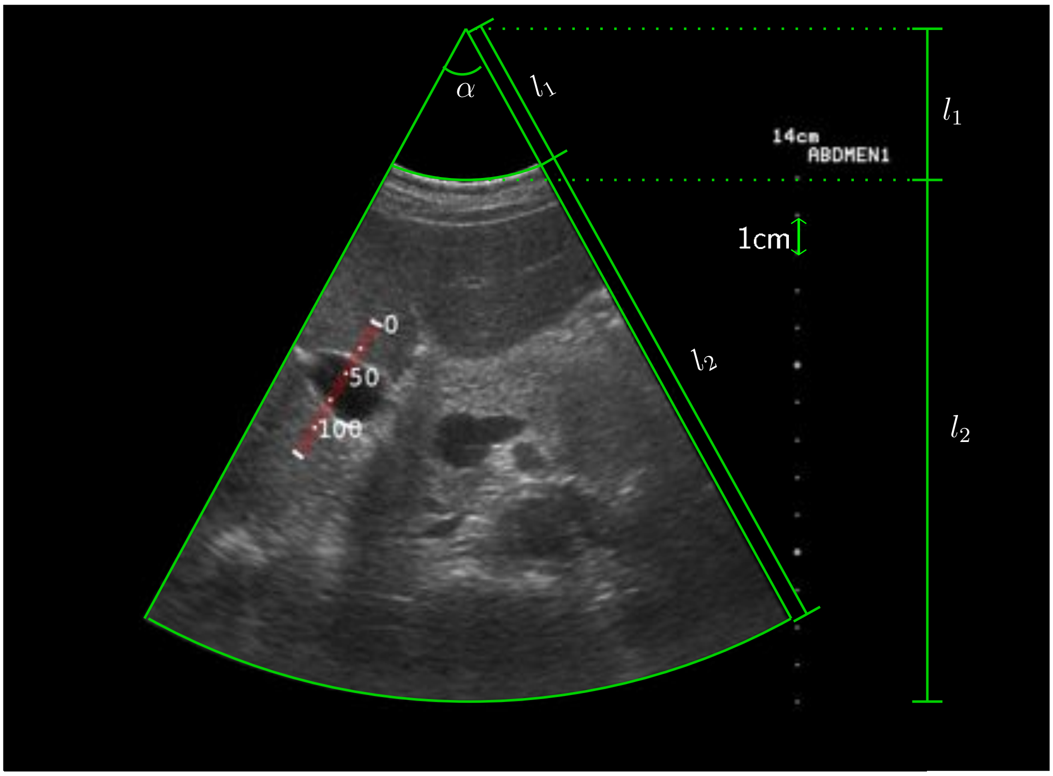


FIG. 4.10: *Ultrasound image using a curvilinear probe – transverse scan of a liver, more medial, with the left lobe (image courtesy of Dr Nicholas Chalmers at Manchester Royal Infirmary). Red line corresponds to the profile in Figure 4.14.*

beam is stored into a binary mask that corresponds to a two-dimensional texture of unsigned bytes (see Figure 4.11). In the mask, the value “0” corresponds to the area of the final MPR image where no information is available, *i.e.* black pixels. The value “255” in the binary mask corresponds to the area in the MPR image where information is available. In practice, the binary mask and the X-ray CT volume dataset can be efficiently used as textures in the rendering stage with multi-texturing. In the fragment program that is used to computed the MPR image, a given fragment is discarded if the corresponding texel of the binary mask is black. Indeed, if we already know that the fragment intensity is going to be null, then there is no need to get the value of the 3D texture that corresponds to the fragment position, and there is also no need to track the needle or eventually insert the lesion. Appendix B.6 show the fragment program that is used to compute the 2D MPR image, to track the needle, and to eventually add a lesion.

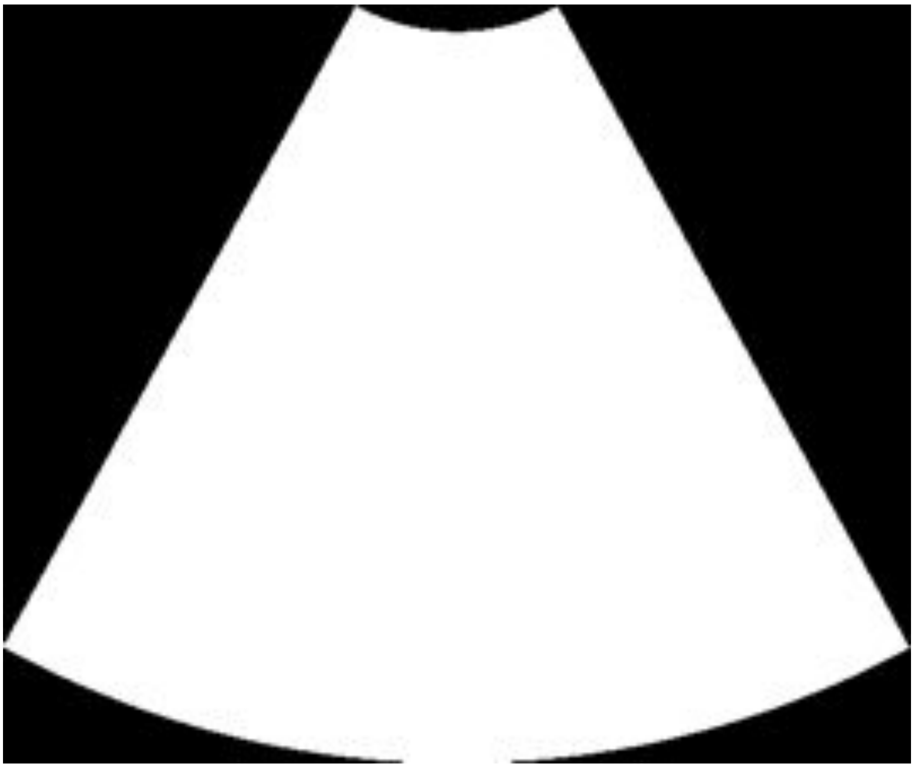


FIG. 4.11: *Example of binary mask used as a 2D texture for computing a MPR image when a curvilinear probe is in use.*

4.6.2 Shadowing in Ultrasound Images

In ultrasound imaging, hard structures such as bones totally reflect the ultrasound signal. The shaft of the needle partially reflects the ultrasound signal. On corresponding images, pixels located behind such structures are therefore in “shadow”. Similarly, pixels located behind gas also appear darker due to sound reflection at the tissue/gas interface. This effect is called “acoustic shadowing”. It can be simulated by post-processing the MPR image to compute a shadow mask (see Figure 4.15b). The shadow mask is a 2D image stored in an array of unsigned bytes. Every pixel of the mask is initialized to the value “255”. To compute the mask, every column is processed as follows: every pixel in the MPR image behind a pixel whose intensity is higher or lower than given thresholds corresponding to bones or gas respectively is set to a smaller value into the mask. Similarly, every pixel in the MPR image behind a pixel whose intensity corresponds to the needle is set to a smaller value into the mask.

Figure 4.12 shows the algorithm that is in used to compute the shadow mask. This algorithm is relatively trivial to implement to compute the shadow mask when a US linear probe is used, *i.e.* when the shape of the ultrasound images to simulate is a rectangle. Indeed, the ultrasound beam can be represented by a set of parallel rays (see Figure 4.13(a)). On the other hand, when a curvilinear probe is used, the rays are not parallel, they describe a circle, which does not match the standard representation of 3D images (see Figure 4.13(b)). Successive pixels of a given “column” (see green arrows in Figure 4.13(b)) are efficiently accessed sequentially using the Bresenham algorithm. This algorithm is generally used to efficiently draw two-dimensional lines. It is relatively fast because the computations are performed using integer numbers only. It would have been possible to use the Cartesian equation of the line for drawing, but computations using floating point numbers are far less effective. Between each iteration, the Bresenham algorithm detects which pixel is the closest of the ideal line. Using edge tracking of the binary mask described in Figure 4.11, it is possible to compute the shadow mask for every “column”. Here, each pixel is accessed sequentially. Such computations cannot be performed efficiently using GLSL because each pixel is treated independently. Consequently, this task has to be performed on the CPU (note that this limitation can now be addressed using Compute Unified Device Architecture (CUDA), a C-like language developed by Nvidia whose first version was released in June 2007). The MPR image is then computed efficiently using the offline rendering of the graphics card. Indeed, using a framebuffer object (FBO) available as a relatively new extension of the OpenGL API, it is possible to render the scene into a framebuffer that is not displayed and that is saved into a 2D texture [192]. After the offline rendering, the MPR image is directly read from this texture or

```
input  : 2D MPR image (input_pixels), bone_threshold, gas_threshold, needle_value,
         bone_shadow_constant, gas_shadow_constant, needle_shadow_constant,
         no_shadow_constant.
output: 2D shadow mask (output_pixels).
begin
    /* Process every column of the MPR image                                  */
    for each column do
        bone_shadow_state ⟵ false
        gas_shadow_state ⟵ false
        needle_shadow_state ⟵ false
        /* Process every pixel of the column                                  */
        for each pixel do
            if input_pixel >= bone_threshold then bone_shadow_state ⟵ true
            if input_pixel is needle_value then needle_shadow_state ⟵ true
            if input_pixel <= gas_threshold then gas_shadow_state ⟵ true

            if bone_shadow_state is true then output_pixel ⟵ bone_shadow_constant
            else if gas_shadow_state is true then output_pixel ⟵ gas_shadow_constant
            else if needle_shadow_state is true then output_pixel ⟵ needle_shadow_constant
            else  output_pixel ⟵ no_shadow_constant
        endfor
    endfor
end
```

FIG. 4.12: *Basic algorithm used to compute a shadowing mask that will be used to simulated ultrasound images.*

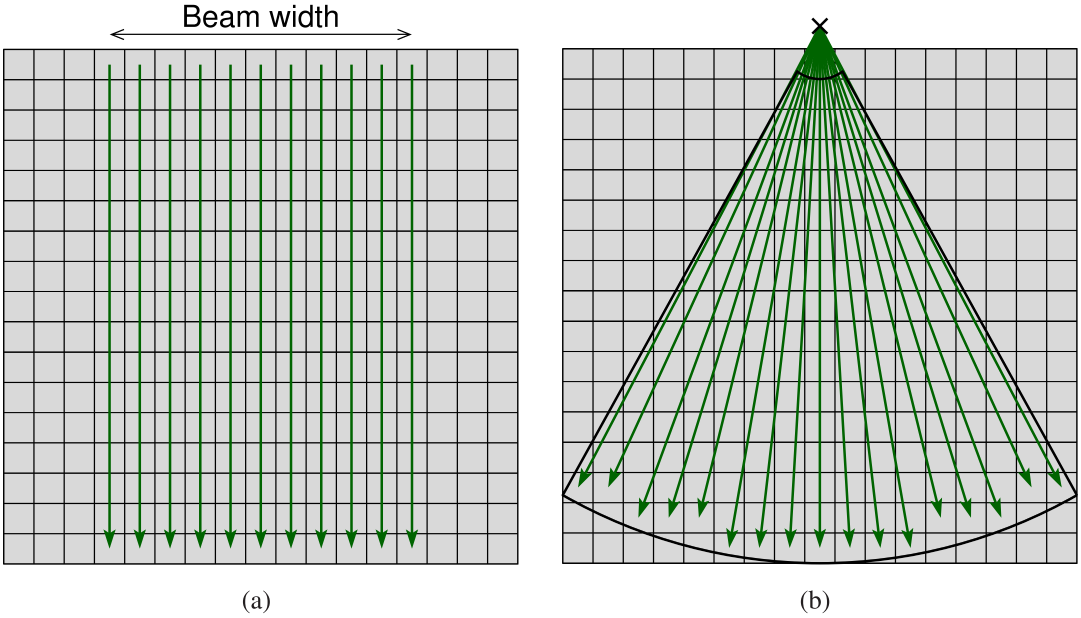


FIG. 4.13: *Pixel access in the case of the use of a) a liner probe, b) a curvilinear probe.*

this framebuffer and the data is transferred into the main memory so that it is accessible by the CPU. This texture will be then used in the final rendering stage to compute a US-like image by multi-texturing using the MPR image and the shadow mask.

4.6.3 Speckle

As it can be seen in real ultrasound images, such as Figures 2.4(b), 2.6(b), 4.10, 6.8(c), and 6.8(d), high-frequency noise is always observed. In our simulation, the noise is represented using a stack of “noisy” 2D slices. This data is loaded into the memory of the graphics card as another 3D texture. A MPR image of noise depending on the probe position and orientation is then computed using hardware acceleration (see Figure 4.15c).

4.6.4 Bright Reflection at Tissue Interfaces

Figure 4.14, a profile of Figure 4.10, shows that bright reflections may also occur in US images at interfaces with bone or gas, but also with fat/tissue and other interfaces. This effect can be produced in the final image by detecting and enhancing horizontal edges in the MPR image. It can be achieved by post-processing the MPR image using a high-pass filter to compute the

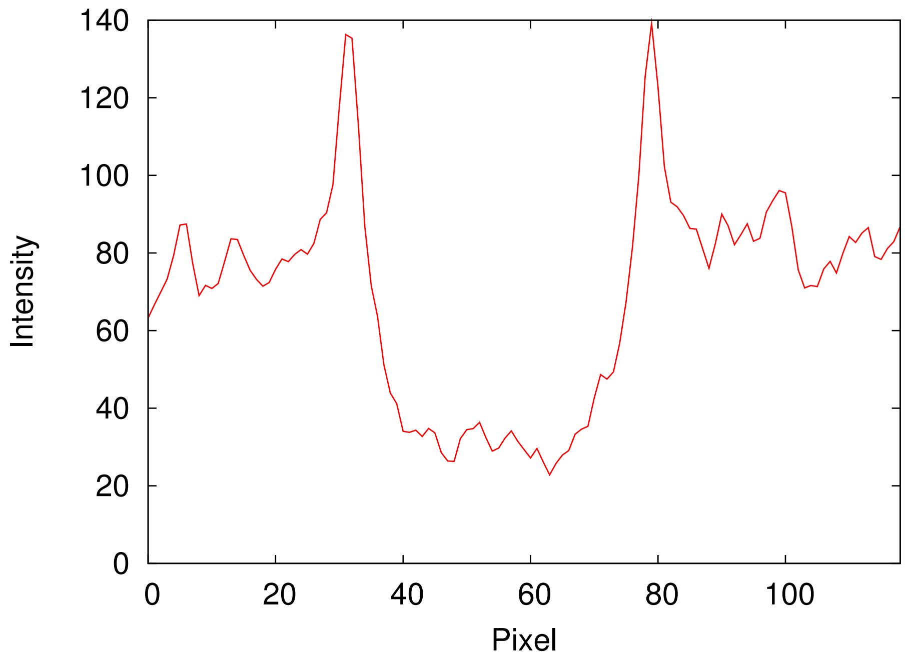


FIG. 4.14: *Profile shows high frequency at interface between two different materials.*

gradient direction (see Figure 4.15e). The Sobel operator has proved to be efficient for such an operation (see the paragraph on **High-pass Filtering** in Section 4.2.1). Reflections will occur when the gradient direction along the vertical axis of the image is above a given threshold.

The patient CT dataset is loaded as a 3D texture of luminance. Similarly, noise data is also loaded as 3D texture of luminance. This is relatively expensive in terms of graphics memory usage. Edge data could also be computed and stored as 3D texture. However, this requires three channels to store the direction of the gradient along the three axis. This will be too costly in term of memory for the now outdated generation of graphics cards that has been used to develop the algorithm. Alternatively, the Sobel filter is applied efficiently in realtime using GLSL.

4.6.5 Image Compositing

In the final rendering stage, all generated images (the MPR image, the shadow mask and the noise slice) are used as 2D textures and blended using multi-texturing. This can be achieved using the fixed rendering pipeline of the graphics card, multiplying the textures together pixel by pixel, as illustrated by Figure 4.15d. However, the fixed rendering pipeline does not provide sufficient flexibility and performance to transform the pixels of the MPR image into an US appearance in realtime via a look up table (LUT). It also prevents the computation of reflections during the rendering stage. Our solution is to efficiently generate these effects (LUT operations, multi-texturing, reflections at interfaces, and also interactive contrast/brightness enhancement) in a single rendering-pass using GLSL (see Figure 4.15e).

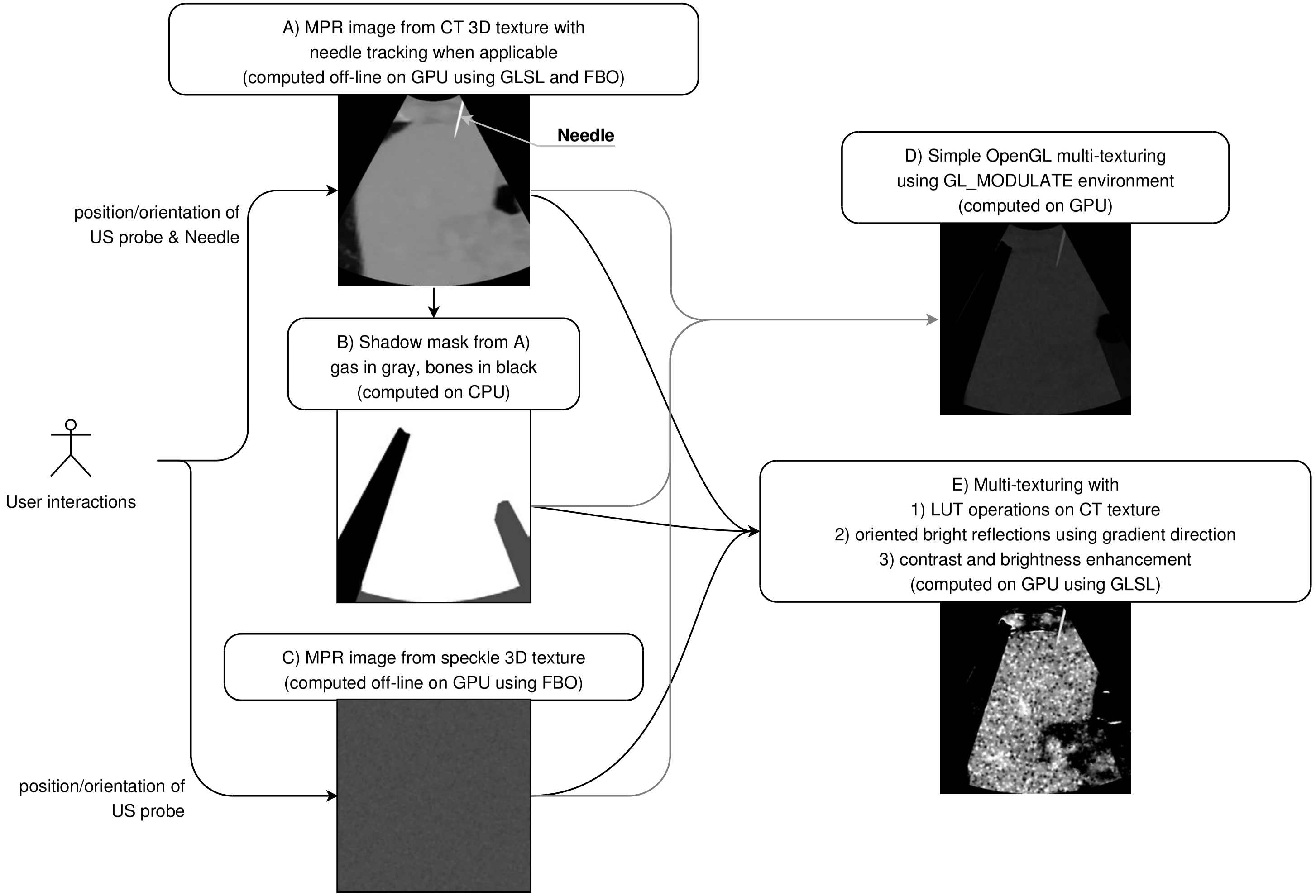

A) MPR image from CT 3D texture with
needle tracking when applicable
(computed off-line on GPU using GLSL and FBO)
Needle
position/orientation of
US probe & Needle
B) Shadow mask from A)
gas in gray, bones in black
(computed on CPU)
User interactions
C) MPR image from speckle 3D texture
(computed off-line on GPU using FBO)
position/orientation of
US probe
D) Simple OpenGL multi-texturing
using GL_MODULATE environment
(computed on GPU)
E) Multi-texturing with
1) LUT operations on CT texture
2) oriented bright reflections using gradient direction
3) contrast and brightness enhancement
(computed on GPU using GLSL)

FIG. 4.15: *Graphics rendering pipeline: a) is the MPR image computed off-line using hardware acceleration and GLSL to represent the needle shaft when appropriate, then LUT operations can be applied on pixel intensities; b) is the shadow-mask computed on the CPU from Image a); c) is the noise inherent to any image modality; d) is rendered by multi-texturing combining the three previous images as textures; e) is the final image after LUT operations, reflections at interfaces, contrast and brightness enhancement.*

4.7 Interactive Determinist X-Ray Simulation on the GPU

4.7.1 Preliminary

The main contribution of the work presented in this section is to show the capability of today's graphics cards to efficiently achieve the physically-based simulation of x-ray images from CT data sets. Note that algorithm speed is an essential concern in the perspective of simulating realistic images at interactive frame rates. This can be used for example to simulate fluoroscopy (see Sections 3.1.1 and 3.2.1 for more details about x-ray imaging). Fluoroscopy is commonly used for guidance in IR procedures, including the guidance of needles.

The simulation of x-ray imaging techniques such as radiography or tomography is being extensively studied in the physics community with applications in crystallography, astronomy, non-destructive testing (NDT) and elsewhere. In these fields of research, real-time simulation is however rarely a requirement. Different physically-based simulation codes are now available and an overview of the physically-based simulation of X-ray images is provided in Section 3.3.7. To resume, there are two different kinds of algorithms for the physically-based simulation of x-rays:

probabilistic method. This method is based on *Monte Carlo* trials. It consists of the individual tracking of each photon during its different interactions with matter [20, 2, 222]. At each step of the simulation, for each photon several events chosen randomly according to physical law probabilities are possible. Obviously such algorithms can produce very accurate x-ray images, but they are computationally intensive.

determinist or analytic method. This method is based on *ray-tracing*. It is possible to optimize calculations by borrowing techniques used in real-time 3D computer graphics, such as Z-buffering, polygon clipping and filling, *etc.* [54, 72, 73].

To compute direct images (i.e. images formed by the x-ray beam transmitted without interaction through the scanned object), determinist methods based on ray-tracing techniques used on CAD or polygon models provide a fast alternative to Monte Carlo methods [73]. However, these realistic simulations of X-ray attenuation are in general the result of a CPU-based approach and are not suitable for an interactive medical application using volume data that must match the acquisition time of a real X-Ray.

It is also possible to adapt such algorithms to simulate x-ray imaging from voxel data, converting each voxel to a polygonized model of a parallelepiped (see Figure 3.7(b)) [263]; but this approach becomes computationally intensive. The simulation of radiographic images from CT data sets has been also reported [111, 144]. More recently, volume rendering using ray-tracing through voxel data has been adapted to the realistic simulation of X-ray [151, 74]. In this case, each voxel corresponds to a cube characterized by its linear attenuation coefficient. From the X-ray source position, a ray is sent through each pixel of the simulated image. Using this approach, the attenuation of the incident X-ray beam is computed for each voxel traversed. Laney *et al* proposed a GPU simulation of X-rays based on volume rendering of unstructured data [141]. Another approach for the simulation of x-ray images from x-ray CT volume data consists of applying the inverse process of the volume dataset creation described in the paragraph on **X-ray Computerized Tomography** in Section 3.2.1 using the Fourier slice theorem [261].

Note that it is also possible to create images that visually look like x-ray radiographs using volume rendering techniques. These methods usually rely on the ray casting algorithm [99, 53] or 3D texture slicing [175]. Ray casting creates an image by casting a ray for each pixel into the volume, and compositing the pixel intensity from a set of samples fetched regularly along the ray (see Figure 3.23); whereas 3D texture slicing consists of slicing the 3D texture with regularly spaced planes parallel to the view plane, the resulting colour is obtained by blending those slices together (see Figure 3.26). More details about ray casting and 3D texture slicing are available in Section 3.3.3 in the paragraph on **Volume Rendering**. These techniques obviously introduce an extra sampling rate: the space between two consecutive samples in the case of ray casting, or the gap between two successive slices in the case of 3D texture slicing. Consequently, they do not directly relate the absorption of light to the properties of each voxel through which the light is travelling. Moreover, introducing a new sampling rate obviously adds an extra possible source of aliasing artefacts. Note that for volume rendering techniques, the original dataset is often reformatted using interpolation so that the voxel size becomes isotropic, *i.e.* voxels are cubes instead of parallelepiped boxes [89].

In contrast, the implementation presented below does not introduce any extra sampling rate or reformatting of the original dataset using interpolation. It uses an efficient ray-tracing implementation on the GPU to compute the x-ray absorption using the attenuation law and takes into account the length of the x-ray path across each voxel crossed by the incident beam. The implementation of our algorithm makes use of a 3D texture to store the volume dataset into the memory of the graphics card and takes advantage of the parallel processing capabilities of this programmable hardware. Indeed, the simulation is well suited for a parallel architecture because the computations for a given pixel are independent from other pixels.

4.7.2 Attenuation Law and Simulation Principles

The basics of the physical properties of x-rays are detailed in Section 3.2.1, therefore only the attenuation law will be recalled here. The attenuation law, or Beer-Lambert law, relates the absorption of light to the properties of the material through which the light is travelling. As stated in the paragraph on **Main Interactions of X-rays with Matter** in Section 3.2.1, only directly transmitted photons are essential for most of x-ray imaging modalities, including fluoroscopy. Then photon scattering will be ignored during the simulation. In the case of a monochromatic incident beam, for homogeneous materials, the Beer-Lambert law is: $N_{out} = N_{in} \times e^{-\mu \times x}$, with N_{in} the number of incident photons (or the input intensity), x the length (in cm) of the x-ray path across the material, μ is the linear attenuation coefficient (in cm^{-1}) of the material, and N_{out} the number of photons transmitted behind the material (or the output intensity). Eq. 3.3 is an expression of the Beer-Lambert law in the case of a monochromatic beam used with a sample made of several materials. Figure 4.16 illustrates in 2D how to use this law in the case of adjacent squares characterized by their own linear attenuation coefficient, such as a tomographic slice. The attenuation of the incident x-ray beam is computed for each pixel traversed. Obviously, in 3D, the attenuation is computed for each voxel traversed. In this case, the voxels correspond to parallelepiped boxes. The next section explains how to pre-process the input voxel data used by our simulator.

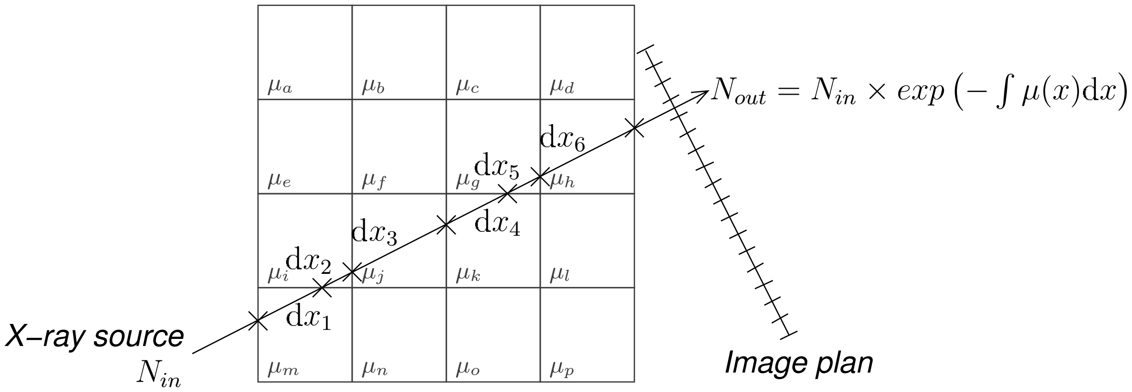


FIG. 4.16: *Principle of the x-ray attenuation by a tomographic slice. The attenuation of the incident x-ray beam is computed for each pixel traversed.*

4.7.3 Pre-processing DICOM Volume Data Sets

The algorithm makes use of patient specific data sets acquired in the DICOM format from a medical CT scanner. Voxel intensities are within the range $[-1000, 1000]$ (Hounsfield scale). Eq. 3.8 shows how a linear attenuation coefficient is converted into Hounsfield units. Hounsfield values are converted into linear attenuation coefficients using:

$$\mu = \mu_w \times \left(1 + \frac{H}{1000}\right) \tag{4.9}$$

with μ_w the linear attenuation coefficient of liquid water at a given energy. As stated in Section 3.2.1, linear or mass attenuation coefficients depend on the energy of the incident beam. The energy typically used for the acquisition of fluoroscopic images of the abdomen is approximatively 80 keV. Using the “XCOM: Photon Cross Sections Database” [17], it is possible to extract mass attenuation coefficients for any chemical element, compound, or mixture, at energies from 1 keV to 100 GeV. It includes water whose well-known chemical formula is H_2O. The mass attenuation coefficient of water at 80 keV given by the XCOM database is 1.84e-1 cm^2.g^{-1}. The density of liquid water ρ_w being about 1 g.cm^{-3}, in our case μ_w can be directly replaced in Eq. 4.9 by 1.84e-1.

Converted values could be directly used by our determinist simulation algorithm based on the attenuation law. However, in the case of a GPU-based implementation, to be accessible by the fragment program, the volume dataset must be stored directly into the memory of the graphics card as a 3D texture. For this purpose, the data has to be rescaled onto the $[0.0, 1.0]$ scale. Figure 4.17 resumes the pre-processing steps of the original DICOM data set in Hounsfield values. Then, the fragment program will compute the corresponding linear attenuation coefficient for each used texel using:

$$\mu(x) = 2\mu_w \times texel(x) \tag{4.10}$$

Note that rescaling linear coefficients between 0.0 and 1.0 is not required in the case of a software only implementation on CPUs.

4.7.4 Ray Tracing

Originally, ray tracing has been designed to synthesize images in the visible domain (see Section 3.3.4). Basically, the principle of ray tracing in 3D space is to compute an infinitesimally thin, straight line of lights from the observer point of view (often called the camera) to each

```
Data: DICOM data set in Hounsfield values ($H$); and $\mu_w$ the linear attenuation coefficient of
      water at a given energy ($\mu_w = 1.84e-1$ in the case of a simulation with a incident beam
      at 80 keV)
Result: texture element (texel) values between 0.0 and 1.0 to be loaded into the texture memory
        of the graphics card
begin
    /* Convert each Hounsfield value ($H$) of the original DICOM data set
       into texel value between 0.0 and 1.0                                    */
    foreach Voxel H of the dataset do
        /* Clamp $H$ between -1000 and 1000                                     */
        if $H < -1000$ then $H \leftarrow -1000$
        else if $H > 1000$ then $H \leftarrow 1000$

        /* Convert $H$ into linear attenuation coefficient ($\mu$)              */
        $\mu \leftarrow \mu_w \times (1.0 + H/1000.0)$

        /* Convert $\mu$ into texel value ($texel$) between 0.0 and 1.0         */
        $texel \leftarrow (\mu - min)/(max - min)$
        /* with $min = \mu_w \times (1.0 + [-1000.0/1000.0]) = 0$, and
           $max = \mu_w \times (1.0 + [1000.0/1000.0]) = 2\mu_w$                */
    endfch
end
```

FIG. 4.17: *From Hounsfield values to texel intensities: pre-processing of the original DICOM data set in Hounsfield values.*

pixel of the projection plane; this line is called a “ray”. In this context, “tracing” means that the intersections of the current ray with every object of the scene need to be computed in order to simulate different physical interactions between the ray and objects. Ray-tracing is widely documented in the literature, and a complete overview can be found in *3D Computer Graphics*, by A. Watt [273].

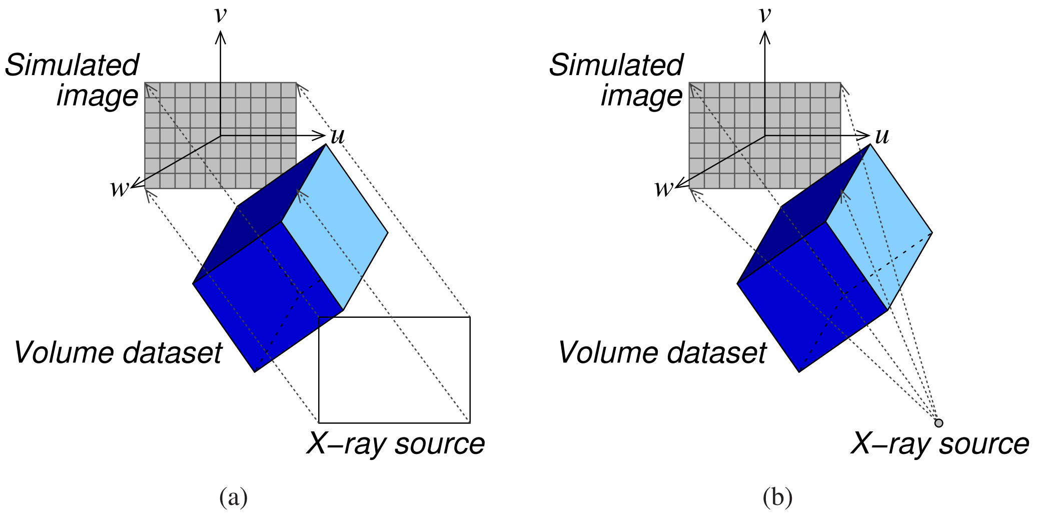
Ray tracing has been shown to be efficient for the determinist computation of x-ray attenuation using CAD models and polygon meshes. Figure 4.16 shows in 2D that the concept can be extended to a cross-sectional map of linear attenuation coefficients, *i.e.* a tomographic slice, each pixel of this plane corresponding to a square characterized by its own linear attenuation coefficient. Consequently, it can also be extended in 3D to CT volume data sets, each voxel corresponding to a parallelepiped box characterized by its linear attenuation coefficient. Parallelepiped boxes are modelled using the equations of three orthogonal pairs of parallel planes. In this context, ray tracing will consist of computing intersections between these planes and a straight line. Appendix B.7 describes in detail how to compute intersections between a box and a ray.

4.7.5 Implementation

We aim to show how to develop a fast simulation code to create real-time fluoroscopic images from patient specific CT data. The incident beam considered in this study is therefore restricted to the monochromatic case using Eq. 3.3. The main program is written in C++ [252] and makes use of the OpenGL ® API [182]. To benefit of the hardware acceleration available on programmable graphics cards and of the parallel architecture of the GPU, the GLSL [213] is used to compute the attenuation of x-rays for each pixel of the image.

Typically, a fragment program is used to compute the final colour of each fragment. In our case the aim of the fragment program is to compute the attenuation of the incident x-ray beam for every pixel. As shown by Figure 3.7, x-ray computerized tomography produces 3D data sets as a stack of trans-axial parallel planes through the body, *i.e.* a discrete volume made of voxels. Consequently, the dataset can be used in the fragment program as a 3D texture. Although the number of voxels and the size of these voxels along the different axes are not isotropic, 3D textures are always considered in OpenGL as isotropic cubes whose edge size is 1.0. Then, to consider each voxel as a parallelepiped box in our ray-tracing algorithm, care must be given in the fragment program to the size of the dataset [in number of voxels] and the size of voxels [in centimetres]. Note that the size of a CT slice is generally 512 × 512 pixels. By contrast, the number of slices of a data is generally smaller than 256. However, because of the limitation of the hardware implementation of graphics cards, the size of a 3D texture in the x, y and z directions, in number of voxels, must be a power-of-two. Empty slices are consequently added to the dataset after the last slice. Obviously, the fragment program will ignore those extra slices during the x-ray attenuation computation.

The implementation of our algorithm makes use of either an orthogonal projection or a perspective projection. The simulated image is orthogonal to the viewing direction. A ray is sent through the centre of each pixel of this image. Figure 4.18 shows the volume dataset with respect to the local space of the simulated image ((u, v, w) coordinate space). For practical reasons, the algorithm is implemented using a “back-to-front” order, *i.e.* the ray is sent from the image plane through the volume. Indeed, in the case of an orthogonal projection, the x-ray source can be defined by the photon direction only. Figure 4.19 shows the simulated image in respect with the local coordinate space of the 3D texture corresponding to the volume dataset ((x,y,z) coordinate space). To be able to use the Beer-Lambert law, we first convert the 2D location of pixels in the screen ((u,v,w) coordinate space, with $w = 0$) to their 3D position with regard to the volume dataset coordinate space in a centimetre scale; indeed the Beer-Lambert law can only be used if distances are in centimetres. Then, as the volume dataset can only be used as a 3D texture, the 3D location of pixels are finally converted into 3D texture coordinates using the same coordinate space as the 3D texture ((x,y,z) coordinate space). During the execution of the fragment program, the location of the intersections of the ray with voxels are computed using the 3D texture coordinate space. They are then rescaled to enable the use of the attenuation law, *i.e.* texture coordinates are multiplied by the size in centimetres of the volume dataset (including extra slices added to get a power of two number of slices).



(a)

(b)

FIG. 4.18: *(a) Orthogonal and (b) perspective projections.*

In practice, the coordinate of every pixel of the simulated image does not need to be specified. Instead, a rectangle (*i.e.* two adjacent triangles) can be displayed using OpenGL functions. The coordinate of each vertex of the rectangle in the (x,y,z) coordinate space is specified using an OpenGL function to assign 3D texture coordinates. The effect is that the fragment processor knows the location of the current fragment (i.e. the rectangle's pixel) with regards to the 3D texture in the (x,y,z) coordinate space. To summarize, the program knows the location of each corner of the image in the 2D screen, and also the location of these corners with respect to the 3D texture. These positions are automatically interpolated for each pixel of the rectangle.

For each displayed pixel of the rectangle, a fragment program will be executed to compute the

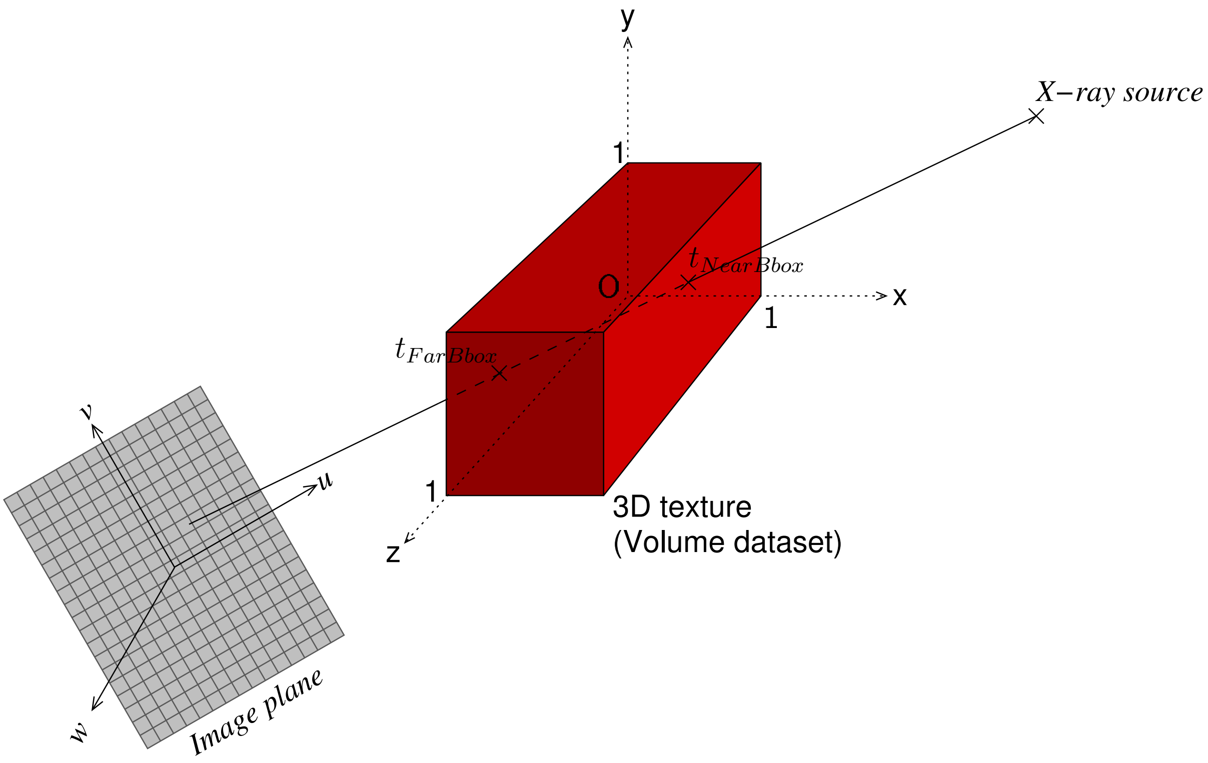


FIG. 4.19: *A pixel of the simulated image can be defined by its local 2D coordinates, or by its 3D coordinates using the same coordinate space as the volume dataset, i.e. the texture coordinate space.*

final intensity of every fragment, *i.e.* the x-ray attenuation. Figure 4.20 is a resume of the algorithm that is implemented in the fragment program. It implements a ray tracing algorithm through the voxels of the 3D texture. For a ray, the first and last voxels to be processed can be detected by computing the intersections of the ray with the bounding-box of the whole dataset. This is used to initialize the ray tracing algorithm when the ray enters into the volume, and to terminate attenuation computations when it leaves the volume. As stated in the previous section, to load the dataset as a 3D texture extra slices are often added at the end of the dataset to get a power-of-two number of slices. To skip these extra slices, the bounding-box does only take into account the size of the original dataset. Our implementation is also optimized by taking into account the fact that the exit-point of the ray into the current voxel is the entry-point of the ray into the next voxel to be processed, which is always within the direct neighbourhood of the current voxel. Indeed, t_{far} of the current voxel becomes t_{near} of the following voxel. Then, only intersections of the ray with the three planes passing through the corner of the current box that is not shared with the previous voxel need to be computed. Given Eq. 3.3, the algorithm can obviously be optimized by computing $-\int \mu dx$ first, and performing the computation of the exponential function at the end. Figures 4.21 and 6.19(b) show x-ray images computed on the graphics card.

4.8 Summary

This chapter has explained how to efficiently model the graphics and haptic properties of the virtual ultrasound probe and of the virtual needle.

As the trainee moves the image probe over the skin of the virtual patient, an image corresponding to the position and orientation of the image plane is computed and displayed in realtime. The goal was to display the image using a variety of formats, including 2D MPR of x-ray computerized tomography (CT), ultrasound-like, and fluoroscopy. This chapter has explained in detail how these images could be efficiently computed from CT volumes in realtime taking advantages of the parallel architecture of modern programmable graphics hardware when applicable.

Motivations for using CT as input data throughout components of the simulator were detailed in Section 4.1. To summarise, x-ray CT is:

- readily available in the hospital;
- undoubtedly the most commonly used modality for 3D data sets;

- the range of voxels' intensity is wide enough to distinguish most organs and tissues.

Iso-surfaces of patients' skin surface have also been extracted from x-ray CT datasets. These are used in the virtual environment for both the haptic rendering and the graphic rendering (see Section 4.2).

To make use of a volume dataset during the graphic rendering stage on the graphics processing unit, the volume dataset must be loaded in the texture memory of the graphics card as a 3D texture. During the needle puncture, the needle has to be tracked as it goes into the virtual patient and the volume dataset needs to be updated accordingly:

- for each voxel within the needle, the original Hounsfield unit is replaced by a high value that corresponds to the metal shaft of the needle;
- between each frame, we must restore the original intensity of every voxel that was located into the needle previously and that are not in the needle anymore.

There is an overhead in transferring the recalculated data from main memory to the graphics card. Using a shading language, we proposed a method to address this overhead (see Section 4.4). In practice, no data is transfered. In a fragment program that makes use of a patient specific 3D texture, before fetching a texel at a given location, we first check if the position is within the needle. In such a case, the texel value is ignored, instead a high value, say one, is used. If the position is outside the needle, then the texel value is fetched as normal. In Section 4.5, we described a similar technique to virtually add a spherical lesion into the patient dataset.

An efficient method to compute a 2D multiplanar reconstruction (MPR) image from the original CT slices was presented in Section 4.3. This method makes use of the texturing capability of the graphics card. This rendering method is also the first stage of our ultrasound-like rendering algorithm presented in Section 4.6. The real-time physics-based simulation of US still remain a challenge. In Section 3.2.2, we saw that speckle noise, acoustic shadowing, specular reflections at the interface of tissues, and enhancement are characteristics of ultrasound images. These artefacts are added to the 2D multiplanar reconstruction (MPR) image to approximate the appearance of an ultrasound image. This can be achieved using the off-line rendering and multi-texturing capabilities of the graphics card. When applicable, computations are performed on the GPU to take advantage of parallel computing. In Section 4.7, we presented a method to simulate x-ray images from volume data in realtime using the Beer-Lambert law on the GPU. It is based on ray-tracing using parallelepiped boxes to model voxels. This method has the advantages to accept non isotropic voxels as input data, and no re-sampling of the data is performed (by contrast to volume rendering using ray-casting).

```
Data: $Intensity_{input}$ (number of input photons), $\mu_w$ (linear attenuation coefficient of water),
      $volume\_dataset\_size$ (in 3D texture units), $3Dtexture$ (volume dataset), $voxel\_size$ (in 3D texture
      units), $gl\_TexCoord[0].stp$ (fragment position with regards to $3Dtexture$), $Xray\_direction$ (with
      regards to $3Dtexture$) (orthogonal projection) or $Xray\_source\_position$ (with regards $3Dtexture$)
      (perspective projection)
Result: fragment colour is the number of photons after attenuation of x-rays through $3Dtexture$
begin
    /* The fragment program is executed for each pixel of the x-ray detector */
    /* Initialisation */
    $Intensity_{output} \leftarrow 0$
    /* Initialize the ray equation, the original of the ray corresponds to
       gl_TexCoord[0].stp, i.e.  the current pixel position with regards to the 3D
       texture */
    $rayOrigin \leftarrow gl\_TexCoord[0].stp$
    if Orthogonal projection then $rayDirection \leftarrow -Xray\_direction$
    else if Perspective projection then $rayDirection \leftarrow$ Normalize($Xray\_source\_position - rayOrigin$)
    /* Compute the lower and upper corners of the dataset bounding box */
    $DatasetlowerCorner \leftarrow vec3(0,0,0)$
    $DatasetupperCorner \leftarrow volume\_dataset\_size$
    /* Compute the intersections between the ray and the dataset bounding box */
    $t_{nearDataset}, t_{farDataset} \leftarrow$ Intersections($ray, DatasetBoundingBox$)
    if there are two intersections then
        $t \leftarrow t_{nearDataset}$
        while $t < t_{farDataset}$ do
            /* Compute the lower and upper corners of the current voxel */
            $Voxel_{lowerCorner} \leftarrow GetCurrentVoxelLowerCorner(t)$
            $Voxel_{upperCorner} \leftarrow Voxel_{lowerCorner} + voxel\_size$
            /* Compute the intersections between the ray and the voxel */
            $t_{near}, t_{far} \leftarrow$ Intersections($ray, voxel$)
            if there are two intersections then
                $path\_length \leftarrow$ ConvertIntoCentimetre($t_{far} - t_{near}$)
                $Intensity_{output} \leftarrow Intensity_{output} + path\_length *$ GetVoxelIntensity($3Dtexture, t$)
                $t \leftarrow t_{far}$
            else $t \leftarrow t_{near} + \varepsilon$
        /* Compute the attenuation using the Beer-Lambert law and convert texel
           intensities into linear attenuation coefficients */
        $Intensity_{output} \leftarrow Intensity_{input} \times \exp(-Intensity_{output} \times 2\mu_w)$
    $fragmentColour \leftarrow$ RGBA($Intensity_{output}, Intensity_{output}, Intensity_{output}, 1$)
end
```

FIG. 4.20: *Computating the x-ray attenuation at a given pixel.*

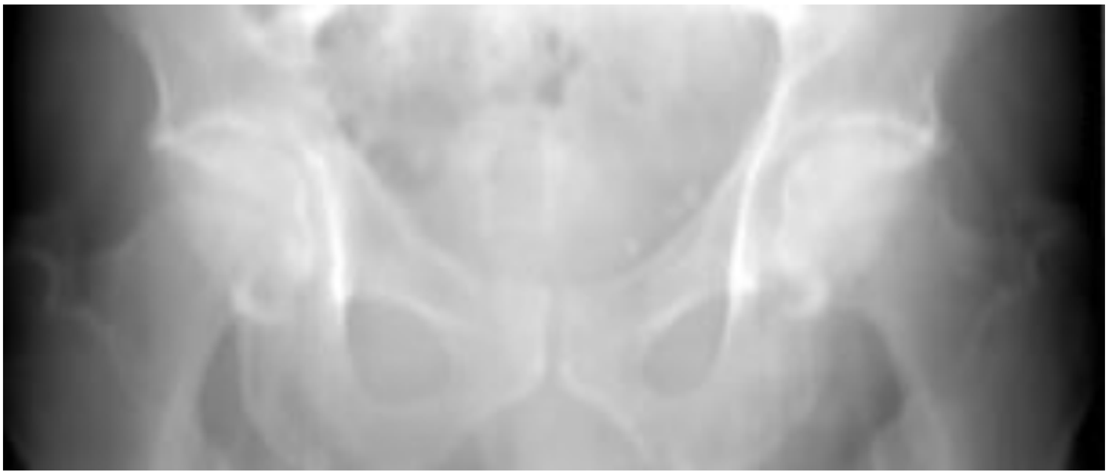
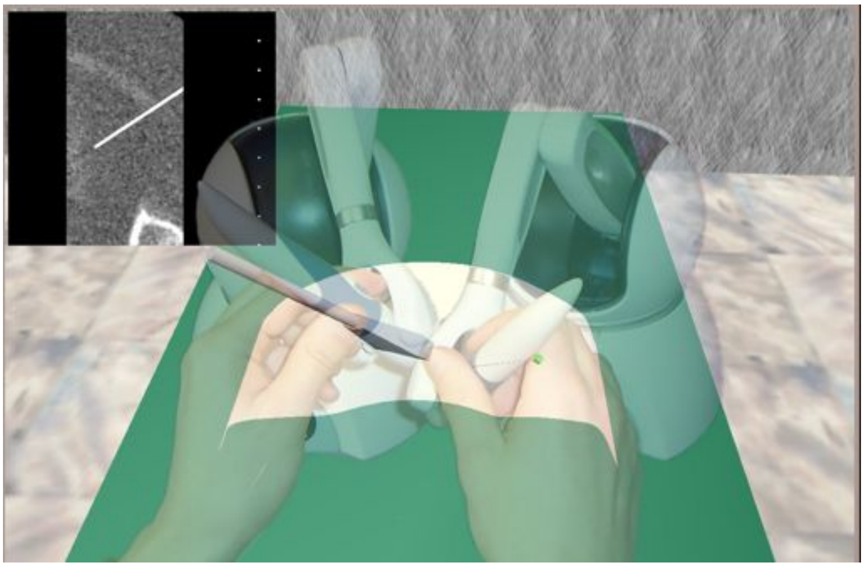


FIG. 4.21: *Example of x-ray simulation.*

Chapter 5

USER INTERACTION WITH THE VIRTUAL ENVIRONMENT



Contents

5.1 Preliminary

The first generation of commercial products implementing a medical virtual environment (VE) focused on laparoscopy where eye/hand coordination and visual sensation are key issues. Whilst VEs have proven to be highly effective training tools for laparoscopy [256], interventional radiology (IR) procedures are far more reliant on the sense of touch [76, 77, 94].

Combining three-dimensional (3D) anatomy models from actual patient data and haptics [218, 246] improves the possibility for the trainees to interact in a natural manner with the virtual environment. Therefore it considerably improves the realism of the simulation and this is a first class requirement for any medical training simulator. In the case of freehand ultrasound guided needle puncture simulation, to interact with the virtual environment, two haptic interfaces are required. Both devices enable the trainees to feel the resistance of the tissue on the needle and the resistance of the skin on the image probe. High fidelity haptic interfaces are particularly important in interventional radiology procedures because the sense of touch and force feedback is also very important in avoiding damage or complications in patients. Nevertheless, the system must also remain cost-effective.

Freehand ultrasound guided needle puncture typically consists of placing a needle into a primarily located target using real-time US images for guidance. Here, the two hands of the clinician are working in unison; to summarize: one hand is inserting the needle whilst the second is trying to locate and display both the needle and the target anatomy on the US images. Obviously, the basic skills required are visuospatial in addition to a well honed sense of touch. Haptics is therefore likely to be of greater importance in attaining “reality” in the simulation of image guided needle puncture procedures, though this is as yet, unproven. Tracking both hands and providing them with convincing haptic feedback are important requirements for an US guided needle puncture training tool.

Section 5.2 describes the PHANTOM® Omni$^{\text{TM}}$ device by SensAble Technologies which has been selected to mimic both the needle and the image probe in our simulator. Section 5.3 presents the haptic models proposed to mimic them. The following section describes how the two haptic devices are positioned and registered within the virtual environment. The chapter ends with a conclusion in Section 5.5.

5.2 PHANToM Omni

After analysing the available options, two PHANToM haptic interfaces from SensAble Technologies have been selected to interact with the VE. Note that the term “finger” is often used in the literature to indicate the position of the haptic device because it substitutes the physical position of the user’s finger. The stylus of the device corresponds to the orientation of the device (see Figure 5.1). The device reads the position of the user’s finger tip and applies a controlled force to it. The device consists of a transmission arm with three direct current (DC) brush motors and encoders. The encoders are tracking the position of the stylus whilst the motors are applying the force feedback. More details can be found in [159, 217, 233].

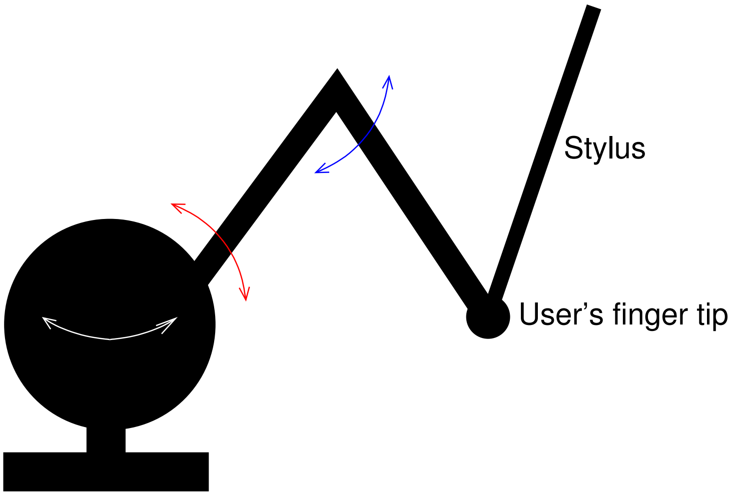


FIG. 5.1: *Schema of the PHANToM device. Arrows represent motors controlling the force feedback.*

The simulator supports most of the PHANToM product range, including the low-cost PHANToM ® Omni ™ (see Figure 5.2) or the more expensive PHANToM ® Desktop ™ (see the picture on page 126). Both devices enable the trainee to feel the resistance of the tissue on the needle and the resistance of the skin on the image probe. BIGN*e*PS*i* supports both left- and right-handed users and allows the trainee to swap the virtual needle and the virtual image probe between the two PHANTON devices as desired. Consequently, the needle and the image probe should ideally be mimicked using the same devices. The PHANToM Desktop provides higher fidelity, stronger forces and lower friction than the PHANToM Omni, but the latter is more cost-effective [234]. For training applications, there is a fundamental trade-off between functionality, accuracy and cost of the devices. Radiologists who tested both PHANToMs with the simulator confirmed that the force rendering using PHANToM Omni devices was close to the real sensations. Thus,

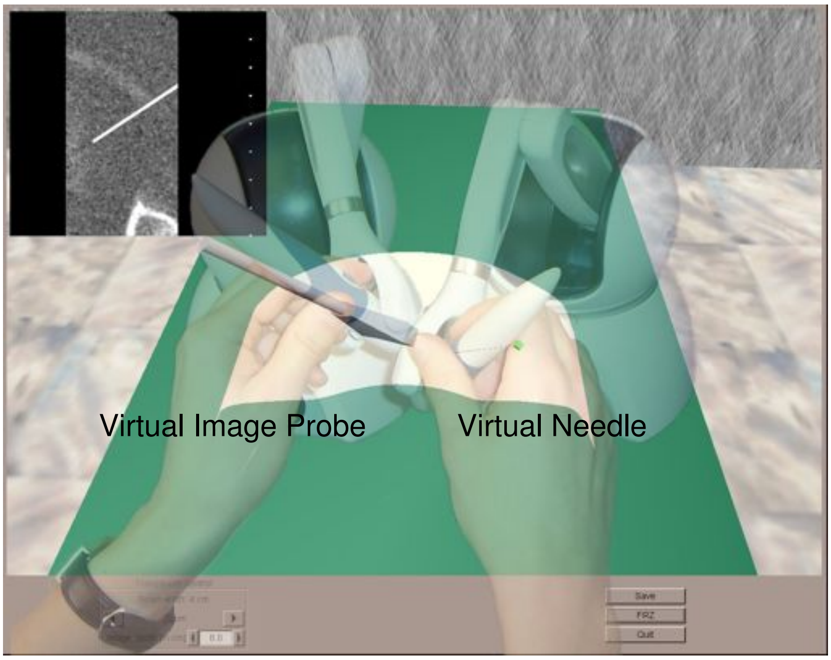


FIG. 5.2: *Screen capture of the early stage virtual environment with the actual haptic devices and the user's hands in background.*

two low-cost PHANToM Omni devices can be used effectively by the simulator. Moreover, to mimic the real experience as closely as possible the stylus on a PHANToM Omni can also be replaced with a physical mock up of a real ultrasound transducer [102], whereas the stylus of the PHANToM Desktop is fixed although tools can be attached to it [236].

5.3 Haptic Rendering

As described above, two haptic devices are attached to the computer. The first acts like a virtual image probe, *e.g.* an ultrasound transducer. As the trainee moves the probe over the surface of the virtual body, he/she is provided with appropriate force feedback so that contact between the probe and the skin can be maintained. The image plane corresponding to the position and orientation of the probe is computed simultaneously using a 3D texture (see Section 4.6) and displayed in realtime in a sub-window of the virtual environment (see Figure 5.2). The second haptic device behaves like a needle and can puncture through the virtual skin and tissue layers. If the needle intersects the image plane of the probe, the 3D texture is updated at interactive rates using the OpenGL ® Shading Language (GLSL) to add the needle so that the trainee will see the needle on the reconstructed image (see Section 4.4).

To achieve acceptable tactile and force feedback perception during the simulation, it is widely accepted that the minimal haptic refresh rate must be 1 kHz [247, 143]. However, refresh rates suitable for real-time graphics must be at least 10-30 Hz. Thus the haptics and graphics renderings are normally disassociated. On mono-processor computers, this can be achieved by using a special timer for haptics, *i.e.* a callback function to be triggered at regular intervals of time (usually in milliseconds). It is also possible to split the program into two or more simultaneously (or pseudo-simultaneously) running tasks by making use of multiple threads. On computers with several processors or now with a single multiple-core monoprocessor, in order to take advantage of the parallel architecture, multiple threads are executed in parallel. In the latter case, care must be given to manage concurrent access to the data written by the haptics thread and read by the graphics thread, and *vice versa*.

The subsections below describe the new haptic rendering algorithms developed to mimic an ultrasound probe and a needle respectively. The first one is a hybrid model which combines i) the proxy-based algorithm on the polygon mesh of the skin surface and ii) a haptic volume rendering algorithm using the CT dataset to include hard structures such as bony landmarks. The second one can be subdivided into two separate models: before penetrating the skin, the needle can be used to explore its surface; after entering the skin, the needle can be used to penetrate internal tissues. Care is also taken to prevent the needle penetrating bones, to match with real world conditions. Using the same surface model in the graphics rendering and haptics rendering make it possible to ensure that contacts between the patient's skin and haptic devices occur exactly at the same time and at the same location. This would not have been possible if the haptic models were only based on volume data and the graphics models on polygonal data.

5.3.1 Ultrasound Probe Haptic Rendering

A proxy-based algorithm [215] is commonly accepted for the haptic rendering of surfaces and this is described in the paragraph on **Force Rendering Algorithms** in Section 3.3.6. The haptic device interacts with the surface of a 3D object, usually represented as a triangular polygon mesh. When scanning the surface of the virtual patient, this model does not natively take into account any internal structures, particularly hard structures such as bony landmarks. However, interventional radiologists often locate the site for probe placement, and needle access using palpation of bone and soft tissue landmarks. There is therefore a requirement to include a similar feature in BIGN*e*PS*i*. It is not at present possible to reproduce the sensation of palpation of the virtual patient using the operator's actual hand. As an alternative, the trainee is able to feel bones through the resistance of the ‘tissues’ beneath the probe mockup. Under these conditions, the force feedback obtained should constrain the device to remain outside the bone.

Using the voxel values in the CT volume dataset, we propose a hybrid model based on the CHAI 3D [40] implementation of the proxy-based algorithm. As Figure 5.3 illustrates, if any hard structure is detected along the straight line between the proxy and the actual position of the device, *i.e.* any voxel intensity between the proxy and the position of the device is higher than a given threshold, then the proxy is moved to the entry point of this straight line into the detected voxel, *i.e.* where the bone is touched. In this case, the stiffness will be increased to simulate a hard object. If no hard structure is detected, then the “classic” proxy-based algorithm is used as normal (see Figure 3.33).

Testing the voxel intensities between the proxy and the position of the device is efficiently implemented by ray tracing through voxel data. In this case, voxels are modelled using axis-aligned parallelepiped boxes made of three pairs of parallel planes [273] (see Sub-section 4.7.4 for details about ray tracing through voxel data). To prevent large force discontinuities between two successive iterations when the device is being moved into hard structures, force smoothing is used [143]. This extension fits in with the “classic” proxy-based algorithm because this model does not introduce any instability or shaking of the device as the resulting force is always parallel to the force that would have been given by the “classic” proxy-based algorithm.

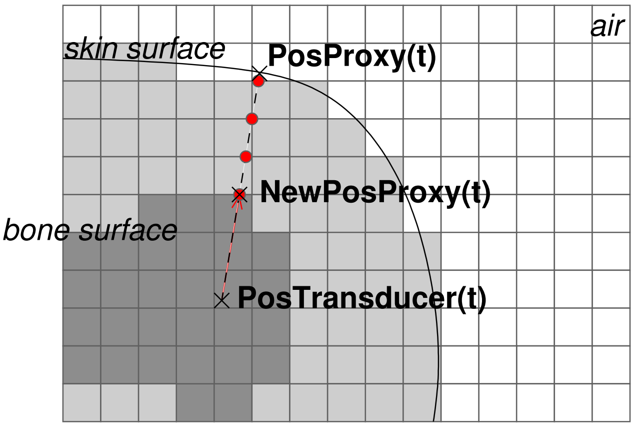


FIG. 5.3: *Force feedback when the transducer is in contact with bones (with* **PosProxy(t)** *the proxy position used normally in the “classic” proxy-based algorithm -* **NewPosProxy(t)** *the position at the surface of bones of the modified proxy -* **PosTransducer(t)** *the device position - all necessary intersections between voxels and a ray from* **PosProxy** *to* **PosTransducer** *in red dots).*

An alternative methods would have been to finely segment hard structures to generate a polygon model of an iso-surface from the volume data (see Section 4.2.2 for details about iso-surfacing). Then it would have been possible to use the proxy-based algorithm with this iso-surface. However, a high number of polygons is required to accurately match bone structures.

5.3.2 Needle Haptic Rendering

Experimental Measurements

The force feedback delivered in VE medical simulator models is generally an approximation to a real procedure, as assessed by experts. Haptics based on real procedural forces should allow a more authentic simulation of the subtle cues perceived when carrying out an interventional procedure in real world conditions [76, 77]. In evaluating needle puncture procedures, for example, *in vitro* studies are essential for detailed understanding of the physical components and effects of overall summated forces. Prior work includes needle insertion forces in porcine and bovine tissues [26, 243, 181].

Our clinical collaborators are collecting such experimental data using a tensile tester used *in vitro* for pig and ox tissues obtained from a Butcher [93]. However, due to the different physical properties of living tissues, *in vitro* data require verification by *in vivo* measurements [193]. Until recently there were few devices available for measurement of instrument forces *in vivo* in humans, unobtrusively: flexible capacitance pads [200] (see Figure 5.4) present a novel opportunity to collect these data *in vivo* and calibration *in vivo* has shown that the output is stable and reproducible [93]. Work has also been performed to estimate both the force required, and the soft tissue deformation during the insertion of a 17 gauge Tuohy needle into a phantom made of a polyvinyl chloride compound [48, 49].

Currently, there is still a shortage of experimental data on tissue resistance. To date, resistance data of the kidney and the liver is available for punctures with different kind of needles. Only data for Chiba needle is of interest here as this is the needle that is often used for gaining access to the kidney, which is a typical example of the visceral interventional needle puncture procedures that we are simulating. Obviously, there is no skin or subcutaneous tissue in the measurements. Figure 5.5 shows the resistance force of a porcine kidney, which is used as the porcine and human anatomies are relatively similar. Figure 5.6 shows a Chiba needle mounted and driven into the tissue at a fixed velocity, 500 mm per minute. This is an approximation of the speed of needle insertion during interventional radiological procedures. The needle orientation was normal to the surface of the kidney. Ten punctures were repeated to obtain an average and a range of forces

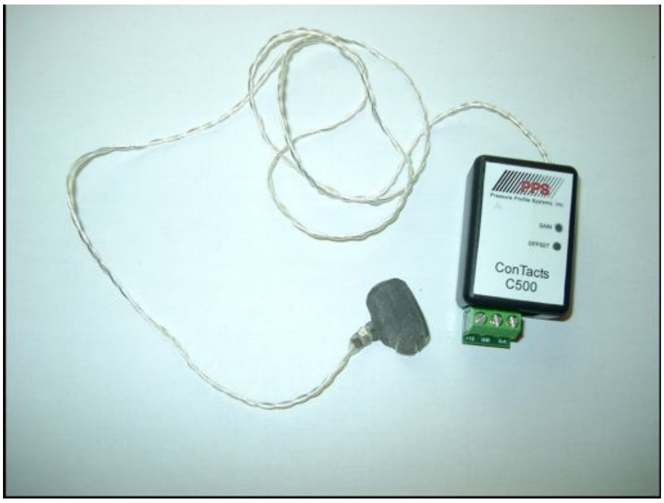


Fig. 5.4: *Capacitance sensor with conditioning unit.*

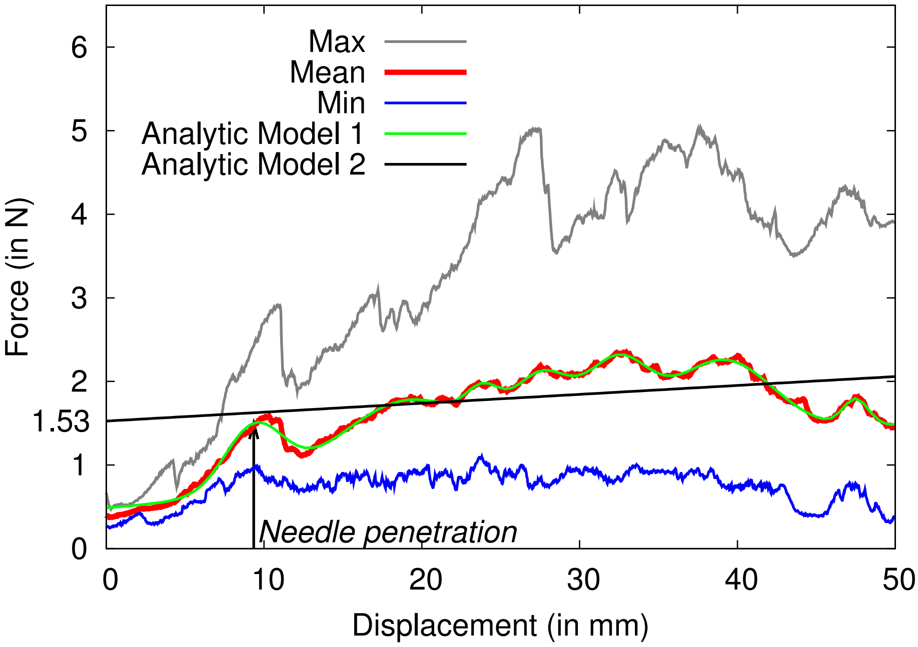


Fig. 5.5: *Force penetration of a Chiba needle in a pig kidney from a tensile tester used in vitro (needle velocity = 500 mm.min^{-1}). Data courtesy of Andy Healey at Royal Liverpool University NHS Trust.*

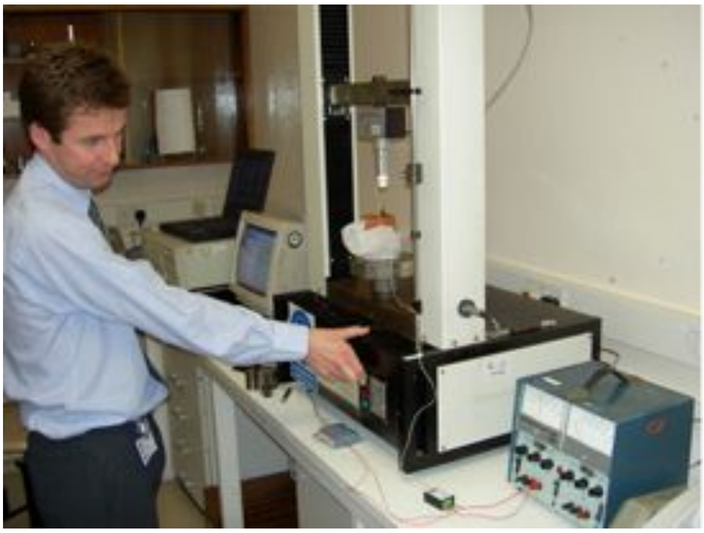


FIG. 5.6: *Experimental set-up, a Chiba needle is mounted and driven into the tissue at 500 mm per minute. Image courtesy of Dr Derek A. Gould at Radiology Department, Royal Liverpool Trust.*

involved. As it can be seen on Figure 5.5, there is an initial displacement, approximately 10 mm, before the penetration of the kidney. Note that this initial displacement is important because in a live case the needle will at first push the kidney away, then a rebound into the force can be observed. It is followed by a plateau phase, although the force slightly increases with the depth of penetration. The black line in Figure 5.5, whose equation is calculated as $f(x) = a \times x + b$, with a equal to 0.01 and b equal to 1.53, is an approximation of the slope of this plateau. The parameters a and b of this equation are estimated using a non-linear least-squares algorithm [6]. As illustrated by the green curve in Figure 5.5, it is also possible to fit an analytic model directly to the measurements. In our case, this model corresponds to Eq. 5.1, a sum of eight Gaussian functions whose parameters are estimated once again using a non-linear least-squares algorithm:

$$\begin{cases} f(x) = \sum_{i=1}^{8} g_i(x) \\ g_k(x) = a_k \times e^{-(x-\mu_k)^2/2\sigma_k^2} \end{cases} \tag{5.1}$$

with $f(x)$ the estimated model, $g_k(x)$ the definition of Gaussian function, a_k the height of the Gaussian peak, μ_k is the position of the centre of the peak, and σ_k the full width at half maximum.

Similar data is also available for bovine livers (see Figure 5.7). Here, there is also an initial displacement before the penetration of the liver, followed by a rebound into the force, then a plateau phase, although the force slightly increases with the depth of penetration. As above, the

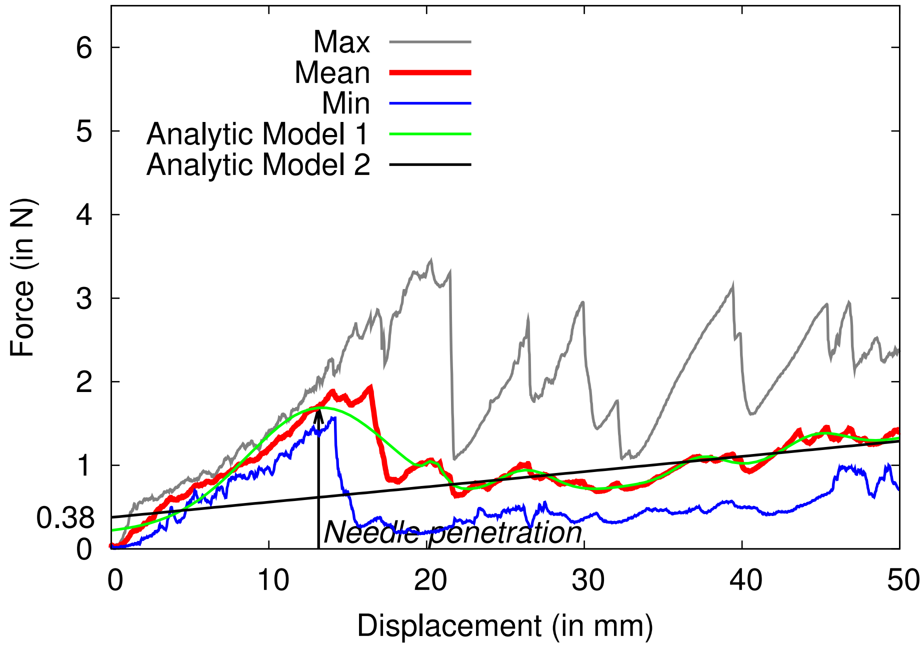


FIG. 5.7: *Force penetration of a Chiba needle in an ox liver from a tensile tester used in vitro (needle velocity = 500 mm.min^{-1}). Data courtesy of Andy Healey at Royal Liverpool University NHS Trust.*

black line in Figure 5.7 is calculated as $f(x) = a \times x + b$, with a equal to 0.02 and b equal to 0.38 (estimated using a non-linear least-squares algorithm). This is an approximation of the slope of this plateau. The green curve is an analytic model fitting the experimental measurements. It corresponds to a sum of six Gaussian functions, whose parameters are also estimated using a non-linear least-squares algorithm. Figure 5.8 shows that to insert the needle into the kidney, more resistance is applied on the needle than for the insertion of the same needle into the liver.

The experimental data presented here cannot be used directly in the simulation as it does not provide a comprehensive description of tissue resistance. Nevertheless, the haptic model described below aims to reproduce the same trend as shown by the experimental data, *i.e.* higher force before the penetration of the tissue followed by a plateau phase. As the slope of the plateau phase is relatively small, the depth of penetration is not taken into account. As shown in Figure 5.8, the average resistance force of the liver and kidney is lower than the maximum force that the PHANToM Omni or the PHANToM Desktop can output, 3.3 N and 7.9 N respectively (see [234]). This is the same for the maximum force resistance of the bovine liver. The maximum resistance force of a porcine kidney is also lower than the maximum force that the PHANToM Desktop can output, but significantly higher than the maximum force of the PHANToM Omni.

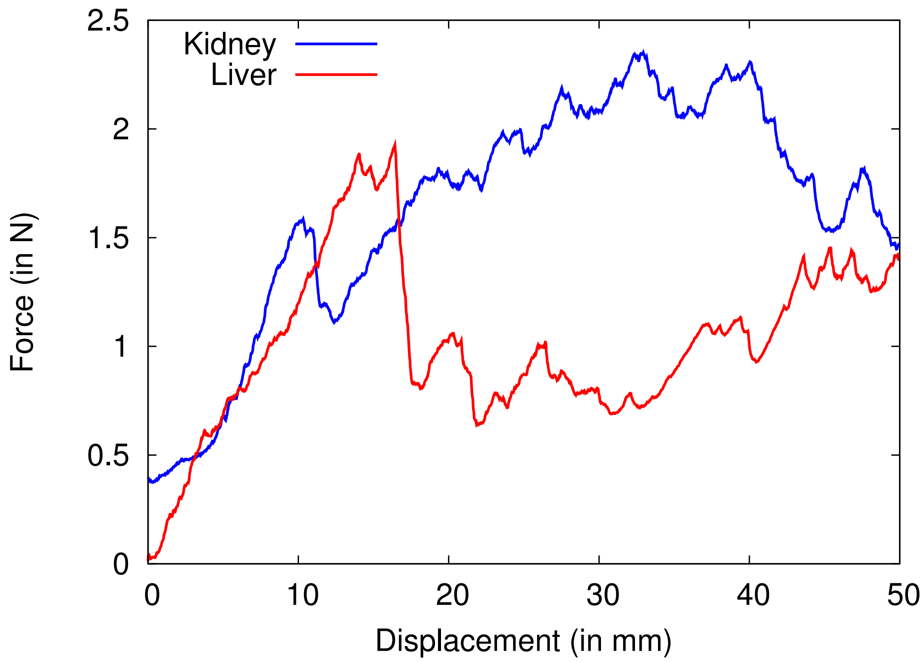


FIG. 5.8: *Comparison of force required for in vitro puncture of kidney, blue trace and liver, red trace using a Chiba needle. X axis is displacement (in mm) and Y axis is force (in Newtons). Image courtesy of Andy Healey at Royal Liverpool University NHS Trust.*

Similarly, the continuous exertable force that these devices can output, *i.e.* during 24 hours, is significantly lower than the tissue resistance (0.88 N in the case of the PHANToM Omni, and 1.75 N for the PHANToM Desktop). However, we are currently investigating support for larger forces with devices that apply force directly to the point of contact through tension in cables, such as the Mantis Workstation (see Figure 3.32(c)) by Mimic Technologies, Inc [170].

Volume Haptic Model

We can distinguish between two different usages of the needle. It can be used for exploring the skin surface before penetration, and it can also be used to penetrate internal tissues. In exploring the surface, normal practice is to first anaesthetise the skin with lignocaine, followed by a small skin incision to facilitate needle entry. The needle is then carefully placed into this skin 'nick', often by gently dragging the tip across the skin until it 'drops' into the skin incision. The needle can then be advanced to penetrate the tissues. During the penetration phase, different features have been identified: higher forces are generated at the interface between two different tissues, with a plateau phase. There is also a very high force when contact with bones occurs. Indeed, the haptic rendering should prevent a needle from penetrating into bones. Our system uses two different haptic models to mimic the needle manipulation. Before penetrating the skin surface, the proxy-based algorithm can again be used, but with different friction parameters to mimic a needle moving over the skin surface. When the force applied by the needle at the insertion point exceeds a given threshold, the haptic rendering mode is switched to a secondary mode that allows the needle to penetrate internal tissues. This can be represented as:

$$\mathbf{F}(\mathbf{t}) = \mathbf{F_1}(\mathbf{t}) + \mathbf{F_2}(\mathbf{t}) + \mathbf{G} \tag{5.2}$$

where $\mathbf{F}(\mathbf{t})$ is the reaction force returned to the haptic device at time t as illustrated by Figures 5.9 and 5.10. t is equal to 0 at the beginning of the insertion. $\mathbf{G}$ is a constant force correcting the effect due to gravity. $\mathbf{F_1}(\mathbf{t})$ and $\mathbf{F_2}(\mathbf{t})$ can be derived as follows:

1. $\mathbf{F_1}(\mathbf{t})$ represents the reaction force to the tissue's resistance as the needle penetrates deeper. Three different sub-models are used to compute $\mathbf{F_1}(\mathbf{t})$ depending whether: a) the needle is penetrating tissues, b) the needle is in contact with bone, or c) the needle is being pulled out.

   (a) Let $\mathbf{PosNeedle}(\mathbf{t})$ be the vector defined by the needle position at time t. When the needle is pushed harder, the resistance of the tissue increases, *i.e.* $|\mathbf{PosNeedle}(\mathbf{t-1})$ $-$ $\mathbf{PosNeedle}(\mathbf{t})|$ will be higher and so the trainee will feel more resistance.

   Let $\mathbf{F_{1a}}(\mathbf{t})$ be the reaction force in a model based on Hooke's law where the reaction effect will be proportional to the depth of penetration of the needle between time $t-1$ and time t (see Figure 5.9):

   $$\mathbf{F_{1a}}(\mathbf{t}) = K_1 \times (\mathbf{PosNeedle}(\mathbf{t-1}) - \mathbf{PosNeedle}(\mathbf{t})) \tag{5.3}$$

   where K_1 is a constant that relates to the tissue properties.

   However, using such a model to calculate the force, the trainee may feel the needle vibrate. To improve stability, the reaction force is averaged using a digital filter over the last m iterations, giving:

   $$\mathbf{F_1}(\mathbf{t}) = \frac{1}{m} \times \sum_{i=t-m}^{t} \mathbf{SubForce}(\mathbf{i}) \tag{5.4}$$

   where

   $$\mathbf{SubForce}(\mathbf{i}) = K_1 \times (\mathbf{PosNeedle}(\mathbf{i-1}) - \mathbf{PosNeedle}(\mathbf{i})) \tag{5.5}$$

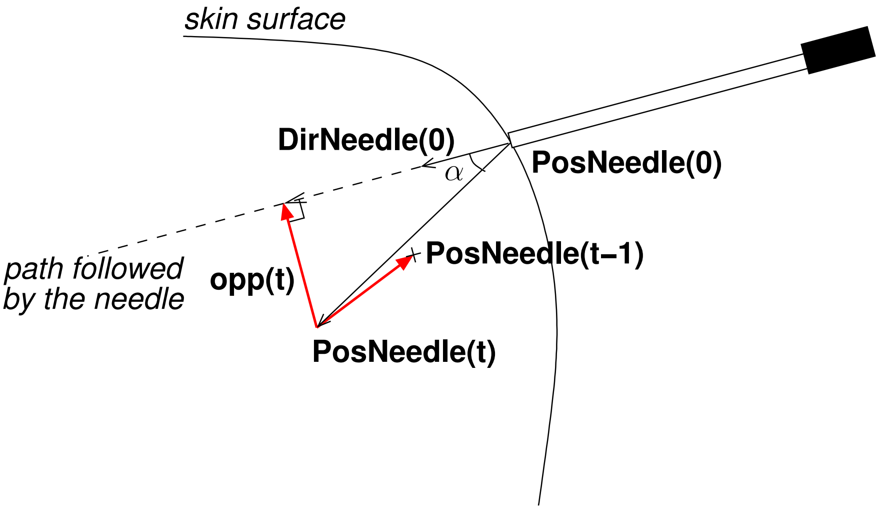


Fig. 5.9: *Forces in action during the insertion of the needle into soft-tissues (with* $\mathbf{PosNeedle(t)}$ *and* $\mathbf{DirNeedle(t)}$ *as the needle position and direction at time t respectively. t is equal to 0 at the beginning of the insertion).*

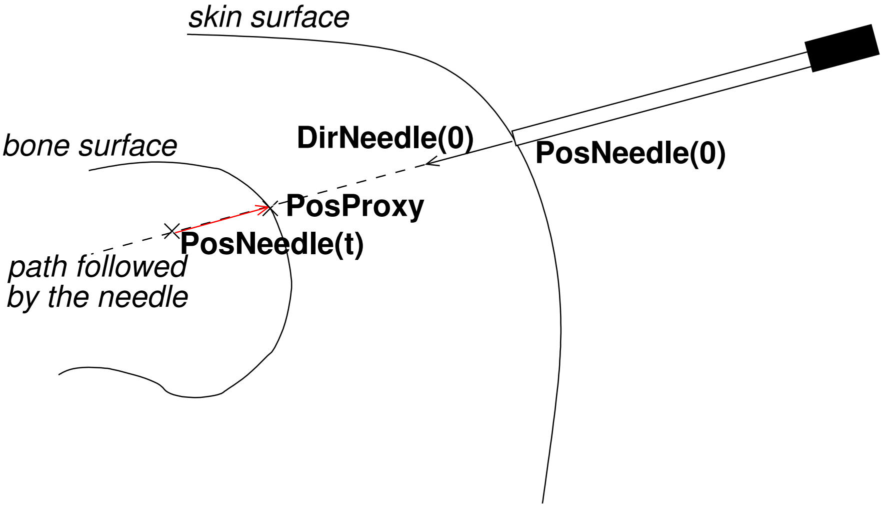


Fig. 5.10: *Forces in action when the needle is in contact with bones (with* $\mathbf{PosNeedle(t)}$ *and* $\mathbf{DirNeedle(t)}$ *as the needle position and direction at time t respectively. t is equal to 0 at the beginning of the insertion.* **PosProxy** *is the last needle position before contact with bones).*

To take into account the tissue properties at the tip of the needle, K_1 is replaced in the final model (Eqs 5.6 and 5.7) by $LUT(VI(\mathbf{PosNeedle}(\mathbf{i})))$, where $VI(\mathbf{PosNeedle}(\mathbf{i}))$ is the voxel intensity at the position of the needle tip in the CT dataset. Tissue properties are thus extracted from a look up table (LUT). To give smoothed variations between the ranges of voxel intensities that characterize tissues, the LUT is represented as a B-spline [198]. Its x-axis is the voxel intensity in standard Hounsfield units, and its y-axis is the resistance coefficient for the given voxel intensity. More resistance is given to hard structures, such as bone, and less resistance to soft tissues. This profile is then manually tuned using experimental force measurements acquired *in vitro* or during procedures on patients in the IR operating theatre [93], *e.g.* more resistance can be given to the range of voxels corresponding to the kidney than to the range of voxels that corresponds to the liver (see Figure 5.8). Obviously, resistance of air, will be very little. Fat is actually a liquid at body temperature that is held in suspension by the tissues so it has very little resistance. Such a haptic LUT is shown in Figure 5.11. The reaction force of the needle to tissue resistance in Eq. 5.4 can be represented as:

$$\begin{aligned}\mathbf{SubForce}(\mathbf{i}) = LUT(VI(\mathbf{PosNeedle}(\mathbf{i})))\times \\ (\mathbf{PosNeedle}(\mathbf{i-1}) - \mathbf{PosNeedle}(\mathbf{i}))\end{aligned} \tag{5.6}$$

Before penetration, as it can be seen on Figure 5.5, or when the needle passes from one tissue type to another, a higher force can be observed [93]. In the final model, this is taken into account by multiplying $LUT(VI(\mathbf{PosNeedle}(\mathbf{i})))$ by a constant value, K_{edge}, when the tip of the needle is positioned at the interface between two tissue types. If the gradient magnitude at this position in the volume dataset is higher than a given threshold, then the needle is located at the interface between the tissues, and the force feedback will be higher. To avoid calculations during the simulation, the gradient magnitude is pre-computed, saved into a file which is loaded when the simulator starts (see Section 4.2). We saw in Section 4.5 that a lesion modelled by a sphere could be artificially added into the patient. When the needle tip is located at the surface of the lesion, it is also at the interface between two tissue types. At an

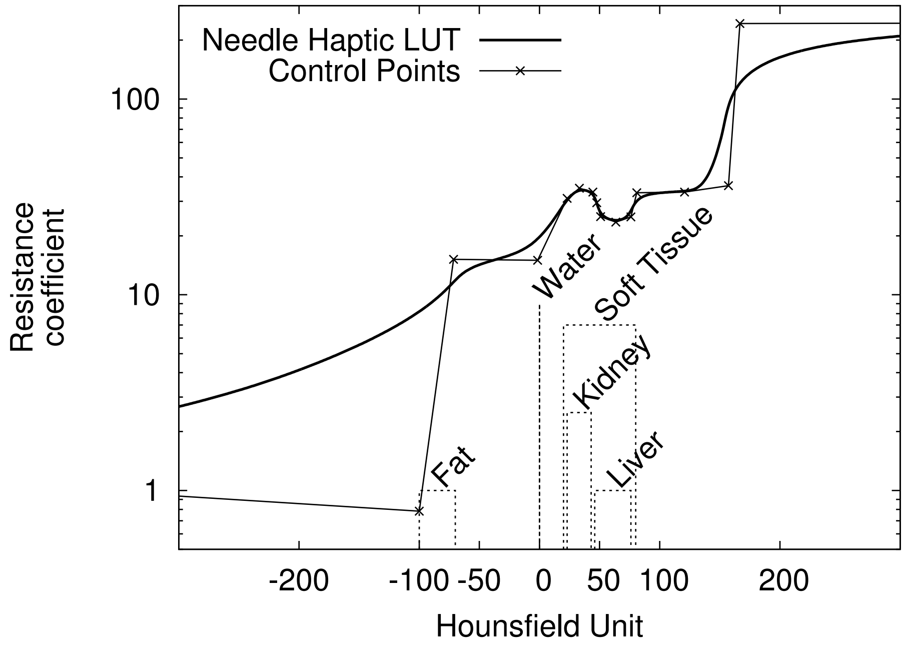


FIG. 5.11: *Haptic LUT for needle puncture. X-axis is the voxel intensity in standard Hounsfield units. Y-axis is the resistance coefficient for the given voxel intensity. The dashed line shows ranges of Hounsfield values for characteristic materials.*

interface, $\mathbf{SubForce}(\mathbf{i})$ becomes in the final model:

$$\mathbf{SubForce}(\mathbf{i}) = K_{edge} \times LUT(VI(\mathbf{PosNeedle}(\mathbf{i}))) \times (\mathbf{PosNeedle}(\mathbf{i}-\mathbf{1}) - \mathbf{PosNeedle}(\mathbf{i})) \tag{5.7}$$

where $K_{edge} > 1$.

(b) The described model (Eq. 5.4, 5.6 & 5.7) gives acceptable results for simulating tissue penetration. In the case of tissue that cannot be penetrated, *e.g.* bones, the model is not well adapted. In such a case the force rendering needed to prevent penetration must be much higher, and the model becomes unstable. To prevent this, when $VI(\mathbf{PosNeedle}(\mathbf{i}))$ corresponds to bones, $\mathbf{F_1}(\mathbf{t})$ is changed to a sub-model similar to the proxy-based algorithm [215]. In the model described in Section 5.3.1, ray tracing through voxel data is used to detect if the US transducer is penetrating bones. The same model can also be used in the case of the needle. However, it is possible to speed up this process taking into account that the needle is following a straight line. Then ray tracing through voxels is not needed to detect the entry point of the needle into bones because voxel values at the tip of the needle can be fetched at each iteration of the haptic loop. Then, in our new sub-model, the virtual proxy represents the position of the device when it enters into the bone (see Figure 5.10). The proxy always remains outside the bone surface whereas the device can slightly penetrate into the bone. The haptic device will be constrained to reach the proxy outside the bone, as illustrated by Figure 5.10, using the following sub-model in Eq. 5.4:

$$\mathbf{SubForce}(\mathbf{i}) = K_{bone} \times (\mathbf{PosProxy} - \mathbf{PosNeedle}(\mathbf{i})) \tag{5.8}$$

where **PosProxy** is the position of the needle at the contact with bones.

(c) When the needle is being removed, the force applied to the needle is relatively constant. Then, if the dot product between $(\mathbf{PosNeedle}(\mathbf{t}) - \mathbf{PosNeedle}(\mathbf{t-1}))$ and $\mathbf{DirNeedle}(\mathbf{0})$ is negative, $LUT(VI(\mathbf{PosNeedle}(\mathbf{i})))$ must be replaced by a constant value in Eq. 5.6. In this case, $\mathbf{SubForce}(\mathbf{i})$ is equivalent to Eq. 5.5.

2. In the real procedure, it is difficult to move the needle off its initial trajectory. In the simulation, once the needle is within the body, it is therefore constrained to follow the path defined by its orientation at the entry point (see Figure 5.9). $\mathbf{F_2}(\mathbf{t})$ corresponds to the required constraining force, which is perpendicular to the needle path:

$$\mathbf{F_2}(\mathbf{t}) = K_2(|\mathbf{F_c}(\mathbf{t})|) \times \mathbf{F_c}(\mathbf{t}) \tag{5.9}$$

where

$$\mathbf{F_c}(\mathbf{t}) = \frac{K_3}{n} \times \sum_{j=t-n}^{t} \mathbf{opp(j)}$$

$$K_2(x) = (1 - gauss(x)) \qquad \forall x, K_2(x) \in [0, 1]$$

K_3 is a constant set empirically in accordance to the desired effect and $gauss(x)$ is a Gaussian function. **opp(t)** can be calculated using basic trigonometric functions on **DirNeedle(0)**, **PosNeedle(0)** and **DirNeedle(t)**, which are known values. Once again, to improve stability, the constraining force is averaged over the last n iterations. To avoid the needle vibrating if it is slightly perturbed from its constrained trajectory, *i.e.* when the distance $|\mathbf{F_c}(\mathbf{t})|$ is negligible, $K_2(x)$ is used to smoothly clamp the constraining force to null.

The PHANToM devices used in the simulation provide 6 DOF in tracking, and 3 DOF in force feedback, so the trainee can move the stylus of the haptic device in all directions and orientations (pitch, roll and yaw), but the force feedback is only applied on the end point of the stylus. A detrimental effect of this is that during the puncture the trainee is able to rotate the pitch and yaw of the needle because the device is not designed to provide any force feedback on rotations. Devices with 6 DOF in force feedback do not have this problem but their significantly higher cost prohibits their use in cost effective medical simulators (see Figure 3.32(a)). To address this drawback, our clinical collaborators recommended not tracking the orientation of the device stylus during the puncture stage. Nevertheless, in actual procedures, after penetrating the skin and whilst the needle is still in superficial tissues, the clinician may have to change the needle angle to successfully reach the target. Consequently the needle orientation is not tracked during the puncture stage unless the needle tip remains 2 cm below the skin surface. This tradeoff does have some impact on the simulator fidelity, however, our clinical collaborators have found the 3 DOF PHANToM to be acceptable.

5.4 Placement of the Devices

BIGN*e*PS*i* has been developed in close collaboration with practicing interventional radiologists. At each stage of development, face and content validation tests have been performed to ensure that the fidelity of the simulator is at an acceptable level. One issue identified was the placement of the haptic interfaces in relation to the virtual patient: placing the Phantoms side by side in front of the computer monitor is not an appropriate mapping to the real situation. The Phantoms both needed to be turned through 45 degrees to give a more natural interface and to avoid potential collisions between the haptic arms. Black crosses in Figure 5.12 illustrate how to position the devices on a template, which is a square whose sides are 156 millimeters.

A calibration program has been implemented to register the relative position of each device. Using the template shown in Figure 5.12, the user moves each device in sequence to five predefined positions (the points **a**, **b**, **c**, **d**, and **e** in Figures 5.12 and 5.13) to estimate the rotation, the translation, and the movement amplitude (using point **f** in Figure 5.13 instead of **e**) of the given device. The current position of the device is recorded when the user clicks on the button of the device. In the simulator, the intersection of the two green lines corresponds to the centre of the world coordinate system, the coordinates in millimetre of points **a**, **b**, **c**, and **d** are (-64.0, 0.0, 0.0), (+64.0, 0.0, 0.0), (0.0, 0.0, -64.0), and (0.0, 0.0, +64.0) respectively. Note that for estimating the movement amplitude of the device, the arm of the haptic device is placed on a calibration box whose fixed hight is known by the program (in this case, the coordinates of point

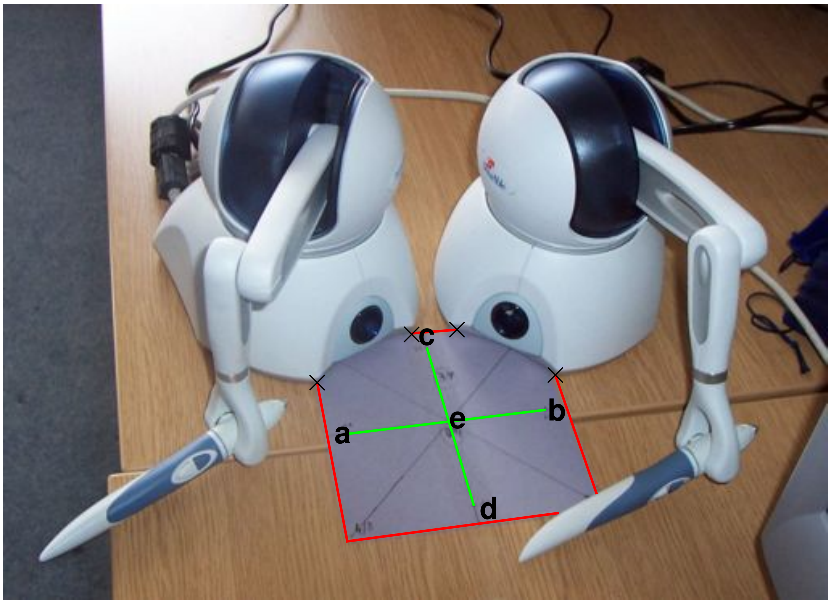


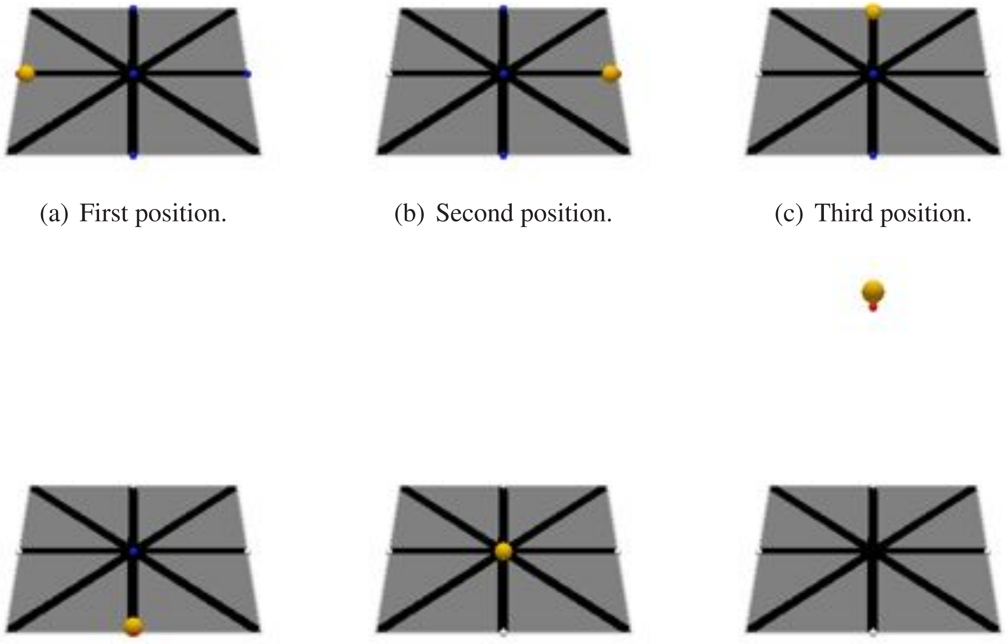
FIG. 5.12: *Positioning haptic devices. The virtual patient is positioned as follows: head above the top of the picture, feet below the bottom of the picture. The monitor displaying the virtual environment is located on the right of the picture. Sides of the red square are 156 millimetres, perpendicular green lines are 128 millimetres.*

f are (x, 143.0, z)). Then, the calibration program will estimate transformation parameters to transform the positions given by the haptic device into the corresponding positions using the world coordinate system.

5.5 Summary

This chapter has explained how to efficiently model the graphics and haptic properties of the virtual ultrasound probe and of the virtual needle.

Section 5.1 detailed choices that have been made to select the haptic devices used by our simulator. One factor has been the high cost associated with purchasing and maintaining the state-of-the-art equipment required. A goal of this project has been to demonstrate that off-the-shelf components can also be utilised to provide a cost effective simulator that still has all of the required functionalities. Specific haptic devices developed for interventional procedures are not suitable for simulating a needle that can be freely moved by the trainee. Indeed, with specific devices dedicated to simulate needle, catheter or guidewire insertions, the position of the needle is restricted to a fixed point and its rotation is restricted. Nevertheless, generic devices can be



(a) First position.

(b) Second position.

(c) Third position.

(d) Fourth position.

(e) Fifth position.

(f) Sixth position.

FIG. 5.13: *Positioning the haptic devices during the calibration.*

freely moved in any position and orientation. Thus they are well suited for mimicking the needle, and also the ultrasound probe. If devices with only 3 DOF in force feedback are suitable for the simulation of a single point probe such as a virtual transducer, these kind of devices are not totally adapted to simulate needles. Indeed, in real patients, once the needle is inserted, it is hard to change its orientation. 6 DOF devices provide a solution but their cost still prohibit their usage in a medical training tool. Consequently, the cost effective PHAMTOM Omni device has been selected to mimic both the ultrasound transducer and the needle. To address the drawback described above, our clinical collaborators recommended not tracking the orientation of the device stylus during the puncture stage.

Section 5.2 described the off-the-shelf haptic device used in the simulation: the PHANToM Omni device developed by SensAble Technologies.

Section 5.3 presented the haptic models proposed to mimic the virtual transducer and the needle. As the trainee moves the probe over the surface of the virtual body, he/she is provided with appropriate force feedback so that contact between the probe and the skin can be maintained. The well known proxy-based algorithm [215] is commonly accepted for the haptic rendering of surfaces but this model only takes into account the surface of objects, not their internal structures. However, as shown in Section 5.3.1, the proxy-based model can be adapted to take into account internal hard structures, such as bony landmarks, using the voxel values in the CT volume dataset. If bones are located between the contact point of the ultrasound probe with the skin and the actual position of the haptic device, the position of the proxy and the stiffness are changed to prevent the haptic device penetrating bones. In Section 5.3.2, the forces required to insert a Chiba needle into a pig kidney and an ox liver were described. There is an initial displacement before the penetration of the kidney or of the liver, then a rebound into the force can be observed. It is followed by a plateau phase. The experimental data can be directly modelled, fitting mathematical functions on the experimental values. Because of the lack of data for various tissues, this model cannot be used. Instead, a volumetric model aiming to reproduced the features described above has been implemented using a LUT and three different sub-models: i) one for the insertion of the needle into soft tissues, ii) one for contacts of the needle with bones, and iii) one if the needle is being removed. The LUT is represented as a B-Spline function, and its control points are tuned according to the experimental force measurements. In order to to accurately tune the LUT and to validate the proposed model, further study will aim to experimentally acquire, using the haptic device with the same experimental apparatus, force measurements when the virtual needle is puncturing a virtual organ to compare the haptic model with the actual forces. Also, resistance force for more kinds of tissue need to be studied. This will enable to replace the coefficients extracted from the haptic LUT by a coefficient computed from analytic models that accurately match the experimental data.

The simulator has been developed in close collaboration with practicing interventional radiologists. At each stage of development, face and content validation tests have been performed to ensure that the fidelity of the simulator is at an acceptable level to allow experienced IR practitioners to test the simulator. One issue identified was the placement of the haptic interfaces in relation to the virtual patient. Section 5.4 described how to position the haptic devices to give a more natural interface and to avoid potential collisions between the haptic arms.

Chapter 6

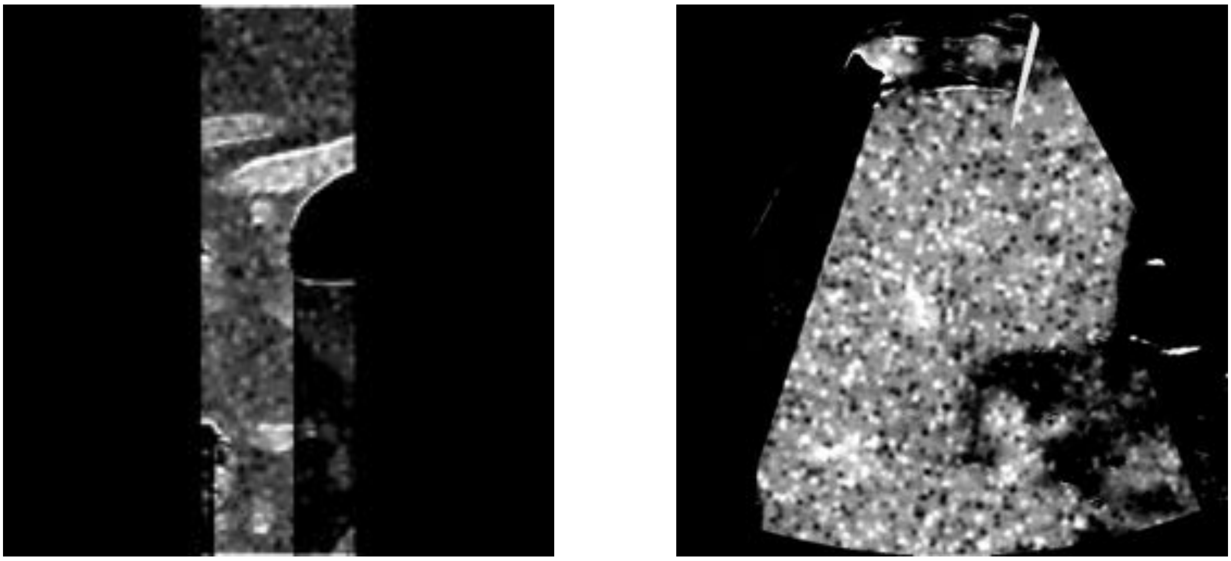
RESULTS

© Photography by Glyn Davies

Contents

6.1 Preliminary

To date, eight DICOM CT datasets containing regions of interest, including the kidney, liver, and neck, have been selected and converted into VRML models. During the simulation, two haptic devices are used. The first mimics an US probe and is used to navigate through the dataset in order to find a suitable cross section image in which to guide a needle for the biopsy. The second haptic joystick is used to simulate the virtual needle. The image computed during the simulation is presented to the operator as a US-like image generated from the CT dataset (see Figure 6.1). The haptic model of the US probe is presented in Section 5.3.1. The needle insertion is simulated using the model model presented in Section 5.3.2 and the location of the needle within the simulated image is represented as a bright reflection, the extent and shape of which conform to the computed needle location and angle through the baseline US image data (see Chapter 4). The validation of the simulator during each phase of the development has been crucial for the success of this research. The section below describes how these components have been integrated to build our ultrasound guided needle puncture training system. Then we present the evaluation of these different components.



(a) US-like image simulated using a linear array probe.

(b) US-like image simulated using a curved array probe.

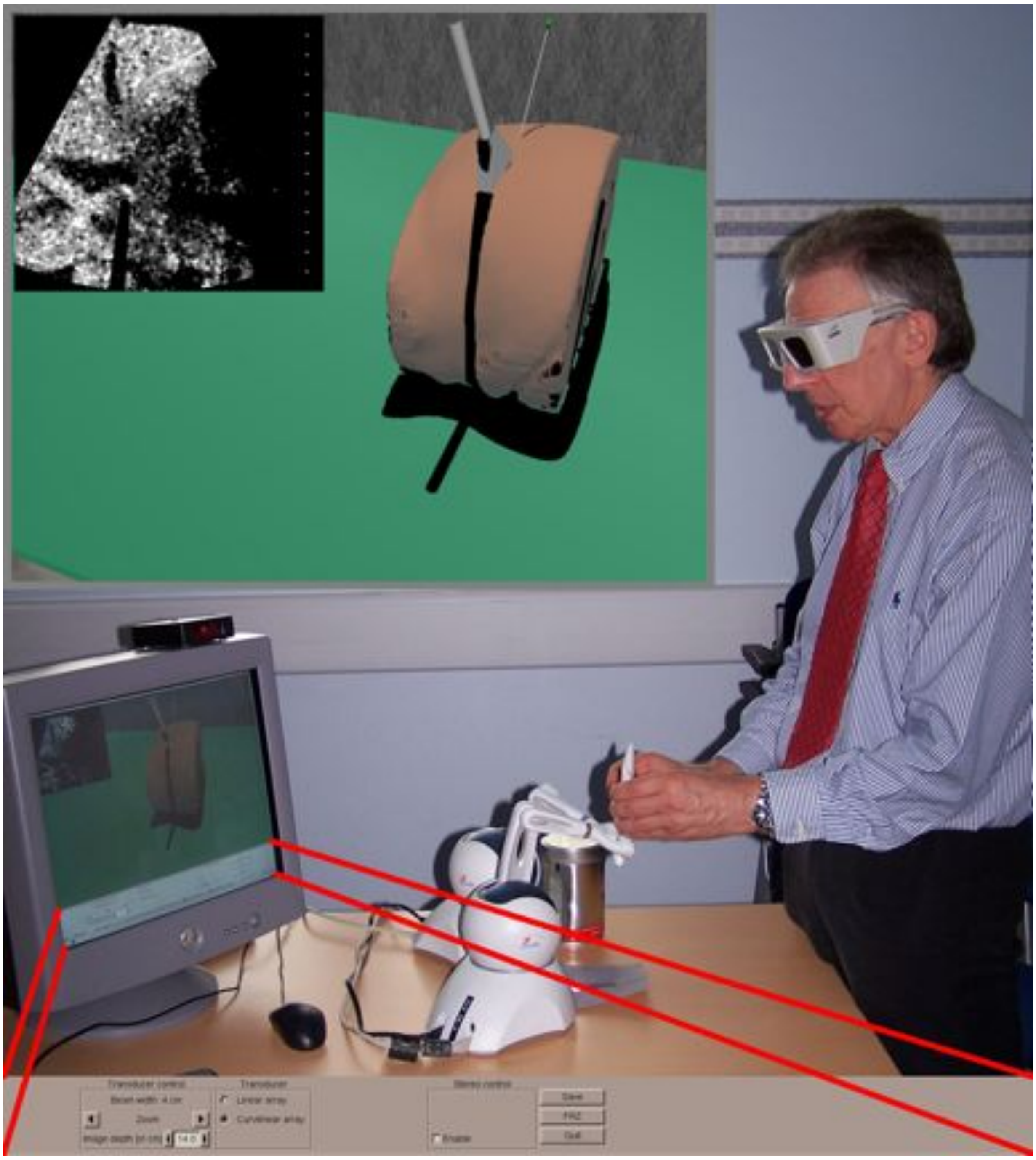
FIG. 6.1: *Simulated ultrasound-like images.*

6.2 BIGNePSi modelling

6.2.1 Software Architecture

Elements presented in previous chapters have been integrated when applicable to build an US guided needle puncture simulator that we called *Bangor Image Guided NEedle Puncture SImulator (BIGNePSi)* [267]. Figure 6.2 shows the integrated simulator being used. The user can interact with the virtual environment via a graphical user interface (GUI) to change typical ultrasound settings such as image depth, select a linear or curvilinear probe, enable or disable stereo, and freeze or save the image. The mouse can also be used to change the contrast of the US-like image. The user can obviously interact using two haptic devices. The first one corresponds to the needle and provides the corresponding force feedback. The second one mimics an US probe and this is used to generate an ultrasound-like image. Figure 6.3 shows the scene graph that has been implemented in BIGN*e*PS*i*. The root node of a scene graph is always the *world node*. This node has three children, the *needle node*, the *US transducer node* and the *interventional room node*. Section 6.2.2 describes the design of the virtual needle and transducer. The interventional room can be modelled using a textured cube to give the appearance of the floor, the walls and the ceiling of a room. This node has two children, a *light node* and a *examination table node*. The light enables illumination to improve the graphics realism. We add an examination table so that the patient does not float in the air but lies of the table. The *patient node* is the only child of the examination table node.

The development of the simulator has been performed on *Windows® XP* and *Linux openSUSE^{TM}*. The programs are written in *C++* [252] and use the *OpenGL® application programming interface (API)* [182] and the *OpenGL® Shading Language* (also called GLSL or GLSLang) [213] for the graphics rendering. The scene management and the haptic rendering has been implemented by extending the functionalities of *Computer Haptics & Active Interfaces (CHAI 3D)* [40]. It is a multi-platform open source library supporting various haptic devices from different manufacturers, and it makes use of OpenGL for the graphics rendering. The GUI is realized using *Fast Light Toolkit (FLTK)* [66]. The *Insight Segmentation and Registration Toolkit (ITK)* [281] is used to automatically pre-process patient specific CT data sets. The *Visualization ToolKit (VTK)* [228] is then used to extract iso-surfaces of the skin of patients from these preprocessed data sets. The iso-surfaces are processed, also using VTK, to reduce the size of the polygon mesh (in term of number of faces) and to smooth the resulting data, and to store the data in Virtual Reality Modeling Language (VRML) files [4].



Beam width: 4 cm

Image depth (in cm) 14.0

○ Linear array

● Curvilinear array

□ Enable

FIG. 6.2: *BIGNePSi in use. The user sees the VE in "3D" using stereoscopy shutter glasses. He is interacting with the VE via two haptic devices, one corresponds to the needle and one one mimics an US probe. This is used to generate an ultrasound-like image corresponding to the position and orientation of the probe. The control panel from the application is shown at the bottom of the figure.*

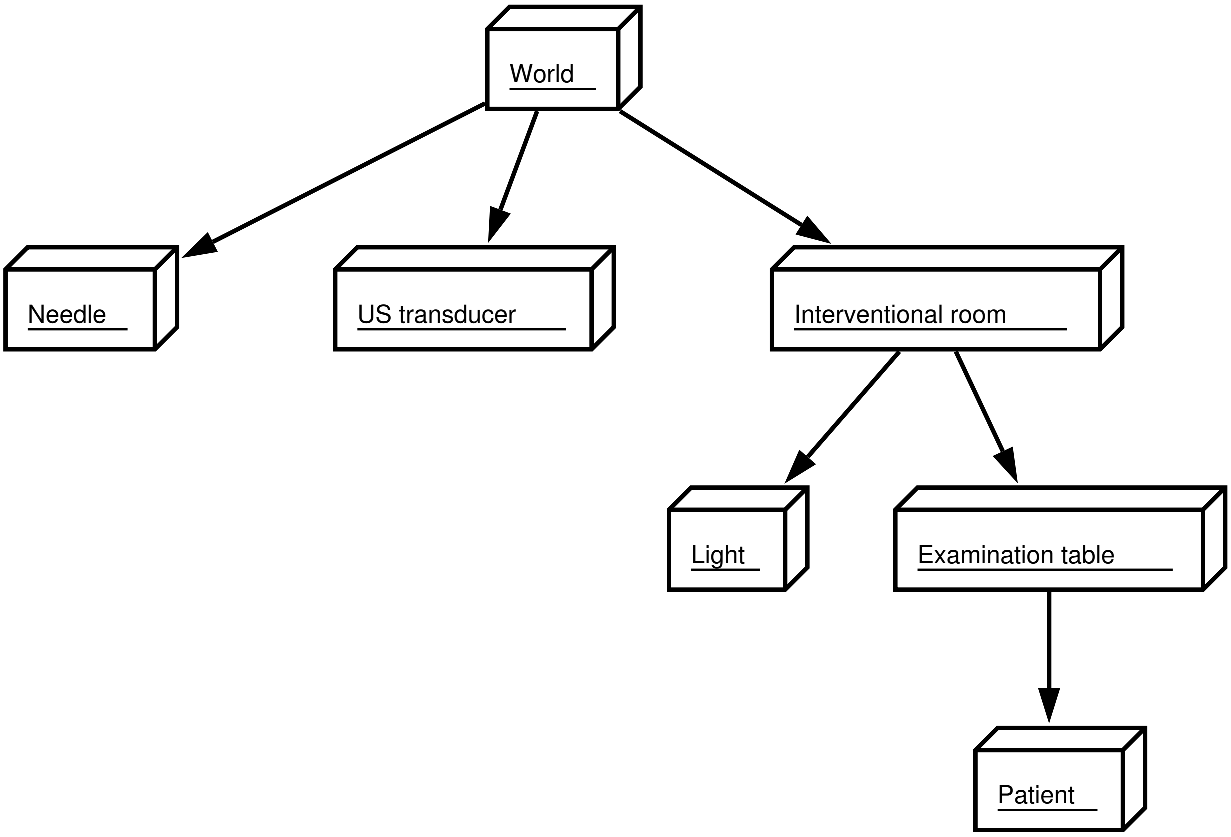


FIG. 6.3: *Scene graph implemented in BIGNePSi.*

A virtual environment should ideally stimulate all five senses of the user. However, today only visual, auditory, and haptic feedback are commonly supported. A basic architecture for this application is shown in Figure 6.4. It can be divided into four main software components [220]:

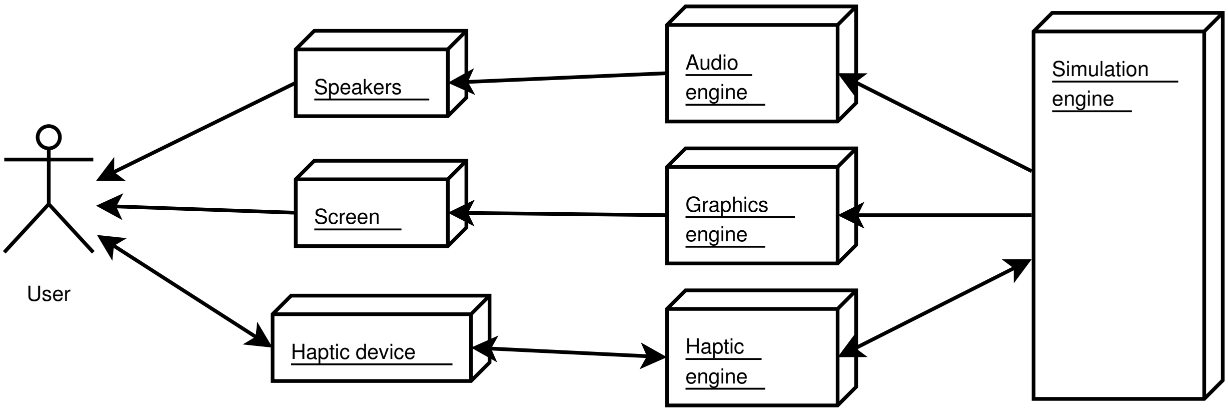


FIG. 6.4: *Main components of a virtual environment simulator incorporating graphics, auditory and haptic feedback.*

Visual renderer: responsible for the computation and display of visual elements;

Auditory renderer: responsible for the audio playback;

Haptic renderer: receives input from the user via the haptic device, computes the response forces, and sends the computed forces to the device;

Simulation engine: synchronization of previous elements over time.

A detailed software architecture of our ultrasound guided needle puncture simulator is given in Figure 6.5. Note that the auditory renderer has not yet been implemented. Such a feature would need to be added to a commercial prototype.

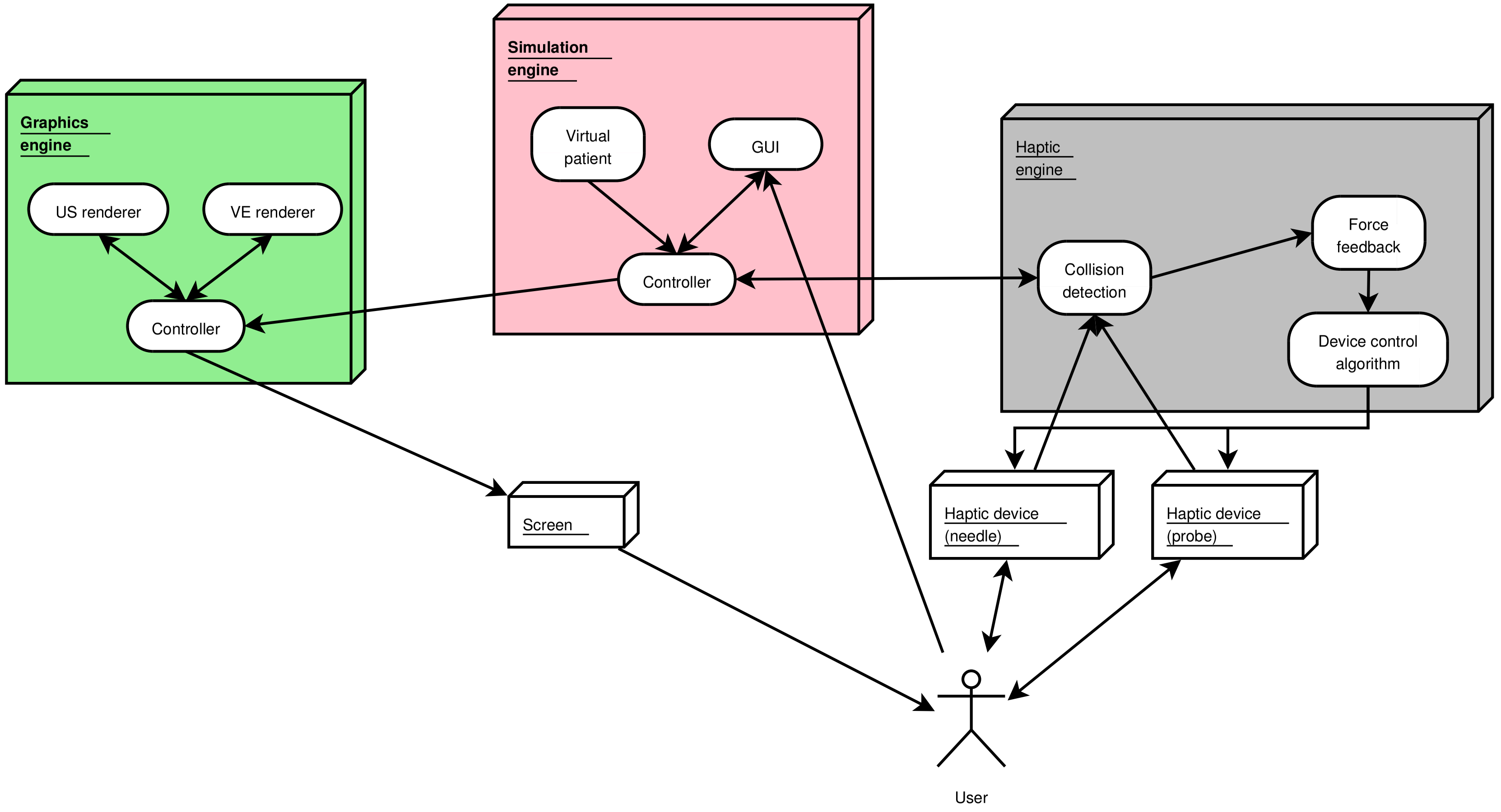
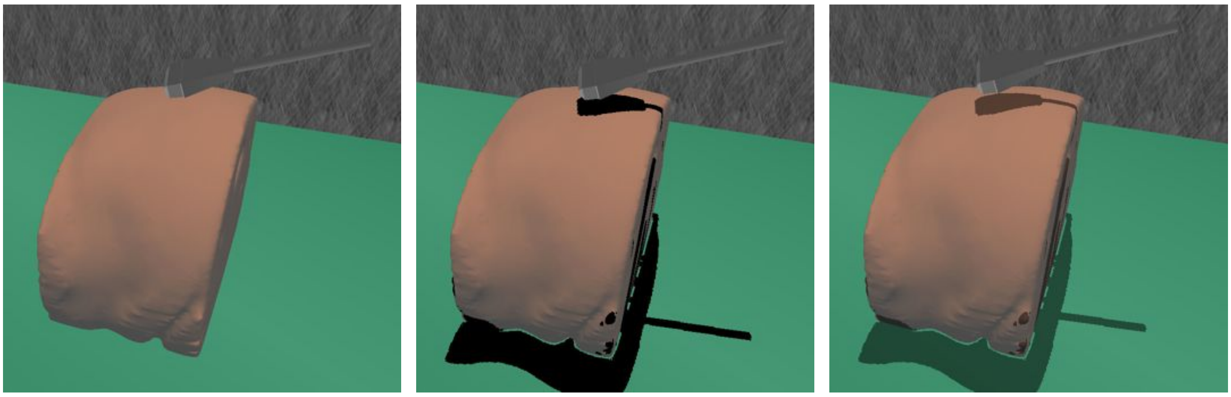


FIG. 6.5: *Detailed architecture for our ultrasound guided needle puncture simulator incorporating graphics and haptics. It is divided into three main components: i) the graphics engine, ii) the hatpic engine, and iii) the simulation engine.*

Graphics engine: responsible for the visual rendering of the real-time ultrasound-like images that are computed depending on the position and orientation of the haptic devices. It is also responsible for rendering the virtual environment. This can be rendered in stereoscopy depending on the capabilities of the graphics card. The virtual environment is made of the patient anatomy, a medical table on which the patient is lying, and a geometric model for each haptic device (see below for a description of the virtual tools). To improve the depth perception, we modified CHAI 3D to support shadowing using the shadow map technique [207, 231, 165]. Figure 6.6 illustrates different rendering schemes. Figure 6.6(a) shows the virtual environment rendered without shadows; and Figure 6.6(b) shows the virtual environment rendered with shadows. If the application detects that the graphics card supports the ARB_shadow_ambient OpenGL extension, then it automatically enables the use of this extension (see Figure 6.6(c)).



(a) Without shadow mapping.

(b) With shadow mapping and without ARB_shadow_ambient OpenGL extension.

(c) With shadow mapping and with ARB_shadow_ambient OpenGL extension.

FIG. 6.6: *Virtual environment with shadow mapping.*

Haptic engine: responsible for the haptic rendering. It is split into three subcomponents. The *collision detection* component detects collisions, *e.g.* between the needle and the patient's anatomy, and between the ultrasound probe and the patient's anatomy. When a collision occurs, a response *force feedback* is computed using the appropriate algorithm presented in Chapter 5. Before sending the response forces to the devices, the *device control algorithms* make sure that the response sent will be suitable for the haptic device, *i.e.* they prevent instabilities and high force discontinuities.

Simulation engine: synchronization of the *graphics engine* and the *haptic engine*. It feeds them with up-to-date shared data, *e.g.* anatomy models, the position of haptic devices, parameters changed via the graphical user interface (GUI), *etc.* To achieve acceptable tactile perception during the simulation, it is widely accepted that the minimal haptic refresh rate must be 1 kHz [247, 143]. However, refresh rates suitable for real-time graphics must be at least 10-30 Hz. Consequently, the haptics and graphics renderings are disassociated. On mono-processor computers, this is achieved by using a special timer for haptics. To ensure a constant refresh rate of 1 kHz, *i.e.* a callback function is triggered at regular intervals of time. On computers with several processors or now with a single multiple-core monoprocessor, in order to take advantage of the parallel architecture, the program can be split into two simultaneously running tasks by making use of multiple threads. Figure 6.7 shows how the flow of events is managed in such a case. Care must be given to manage concurrent access to the data written by the haptics thread and read by the graphics thread. Indeed, multiple threads are executed in parallel. When a thread is currently writing shared data, it should block the other thread from reading that data. If not, the data that will be read may be invalid. Similarly, when a thread is currently reading shared data, it should block the other thread of over-writing that data. If not, the data that is being read may become invalid. The management of concurrent access can be achieved using mutex [86].

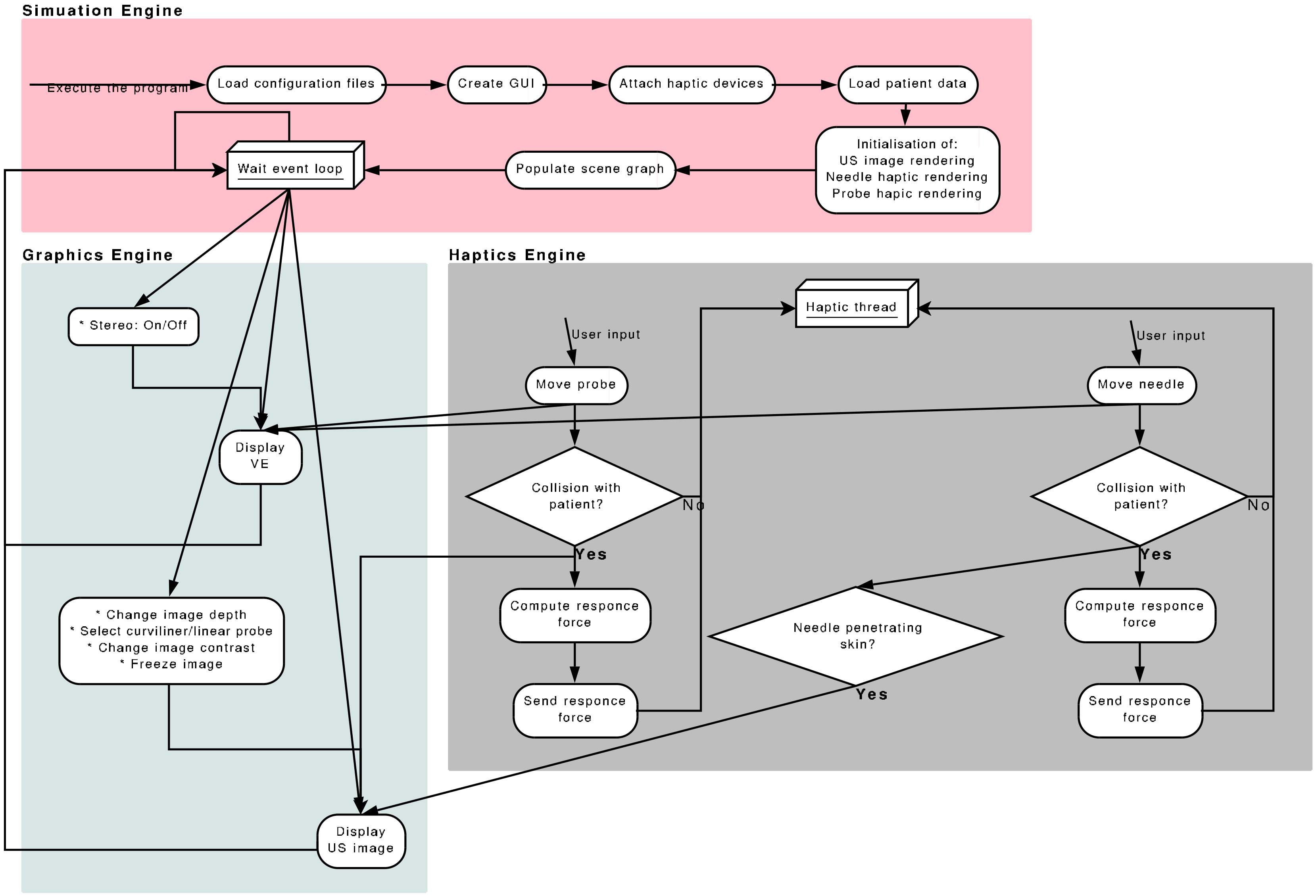


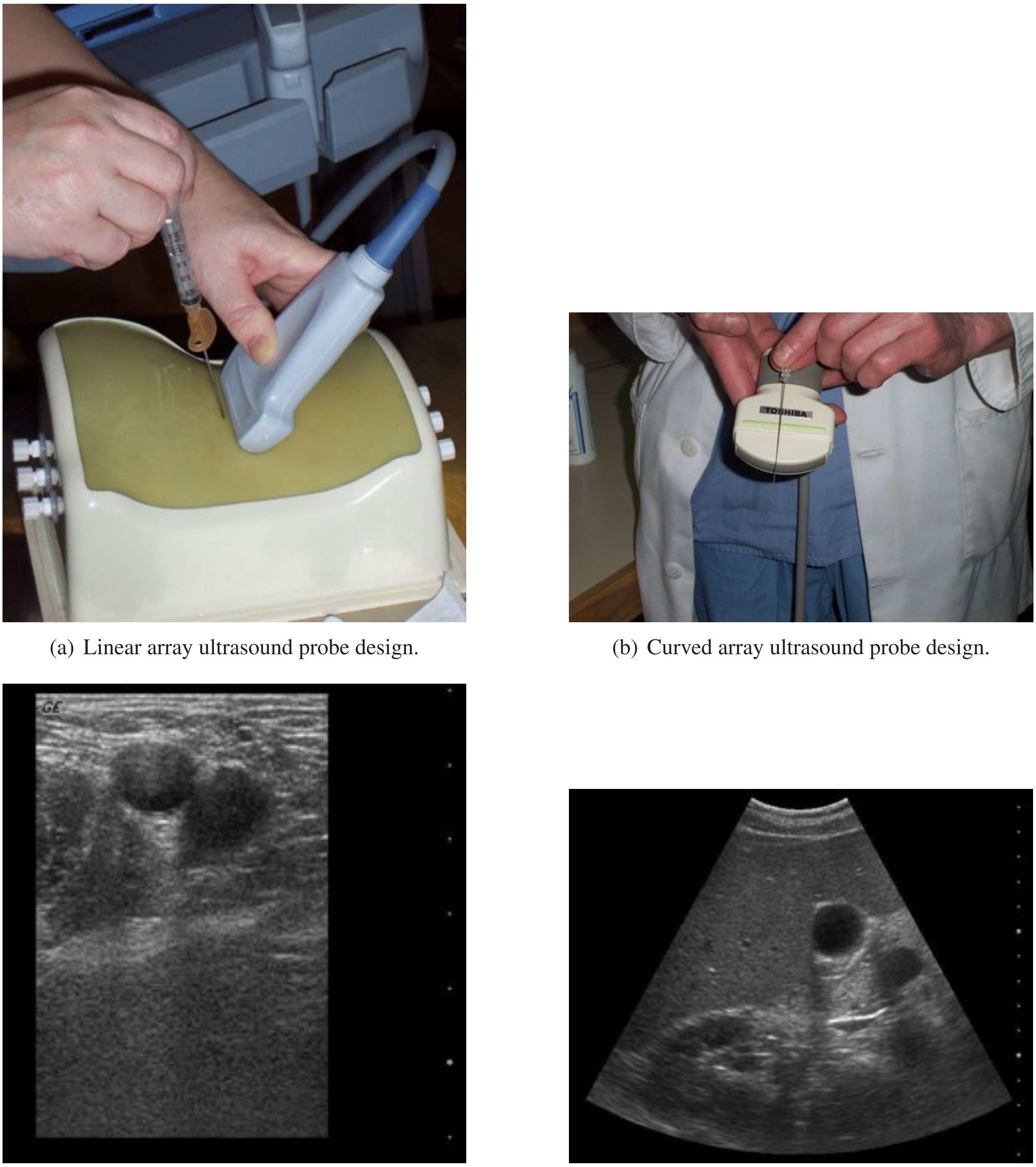
FIG. 6.7: *Event management for our ultrasound guided needle puncture simulator incorporating graphics and haptics in separate threads.*

6.2.2 Description of the Virtual Tools

Ultrasound Probe

Ultrasound probes (called transducers) are used in medicine to generate, detect and measure directional acoustic fields (or beams). To achieve this, arrays of piezoelectric crystals are generally used. When an electric voltage is applied on piezoelectric crystals, they change size. Applying an alternating current (AC) will make them to oscillate at very high frequencies, consequently they will generate very high frequency sound waves, *i.e.* ultrasounds. Additionally, they generate a voltage when a force is applied to them (this phenomena is called "piezoelectricity"). Thus the same piezoelectric crystals are also used to detect and measure US echoes. For IR procedures, the acoustic beam may have a circular or rectangular aperture produced by a curvilinear array probe or a linear array probe respectively. As illustrated by the pictures in Figure 6.8, the curvilinear array probes just angle the detectors outwards while the linear probes are straight. There are usually 128 piezoelectric elements in both the linear and curvilinear probes as a standard. The more elements there are, the more expensive the transducer is, but not by any fixed number or ratio. The spacing is equal between elements. On the linear probe commonly used by our clinical collaborators, the piezoelectric element array measures approximately 5 cm × 0.8 cm; on their curvilinear probe it is 10 cm × 1.2 cm, but these dimensions will vary from design to design.

As can be seen from Figure 6.8, a medical ultrasound transducer has a complex shape. The virtual ultrasound transducer used in the simulator has been created using *CosmoWorlds* and uses the VRML file format. VRML 97 is the established ISO standard used on the Web to describe three-dimensional data [4]. For real-time applications, usually a reasonably small number of polygons is commonly used for 3D objects that do not require high quality rendering. Figure 6.9 shows the virtual transducer as used in the simulator; even with a few polygons, 126 triangles, the visual quality of the US probe is reasonable. The simulator reads this VRML file using a class provided by the Visualization ToolKit (VTK) [228], *vtkVRMLImporter*, and converts the data and its visual properties into corresponding instances of an object class provided by CHAI 3D [40] for enabling the tool to use haptics and for its graphical renderings.



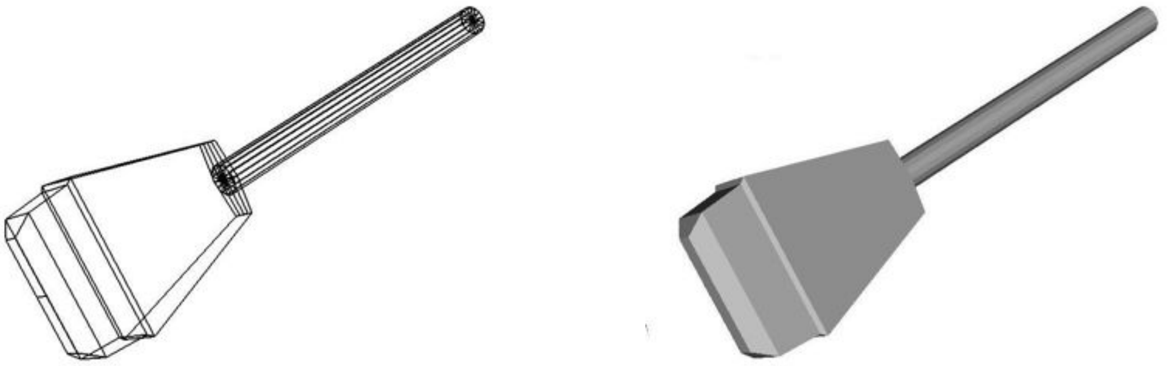
(a) Linear array ultrasound probe design.

(b) Curved array ultrasound probe design.

(c) Example of ultrasound image with a rectangular aperture: traverse groin (artery and vein, bone behind). Image courtesy of Dr Nicholas Chalmers at Manchester Royal Infirmary.

(d) Example of ultrasound image with a circular aperture: transverse liver. Image courtesy of Dr Nicholas Chalmers at Manchester Royal Infirmary.

FIG. 6.8: *Pictures of real medical US probes, and examples of corresponding images.*



(a) Wireframe rendering using Octaga Player [180].

(b) Gouraud shading rendered with OpenGL using Octaga Player.

FIG. 6.9: *Pictures of the virtual transducer used in the simulator.*

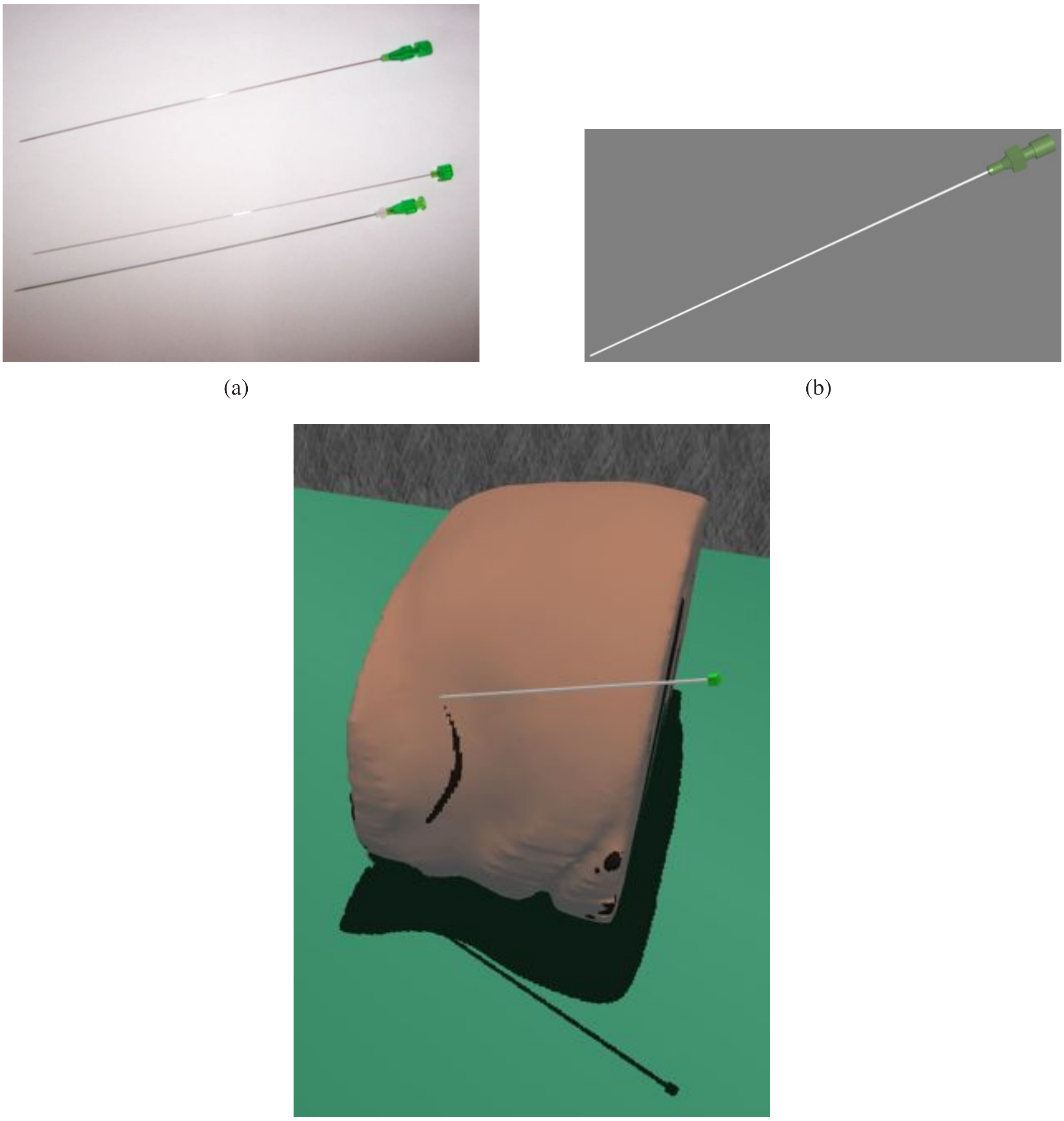
Needle

Different needles, usually named after the companies that initially produced them, have different tip designs. Some are designed to cut, some to steer and some to push aside or separate tissue. Needle calibre in medicine is measured in gauge, the higher the gauge number the smaller the calibre. To get access to the kidney, Chiba needles are often used. Figure 6.10(a) shows a real co-axial Chiba needle whose gauge is 21 and is length 21 cm. Co-axial needles are designed to pass another needle through their centre.

It is possible to accurately reproduce such a needle using a 3D content creation package such as *CosmoWorlds* (see Figure 6.10(b)). However, due to the relative simplicity of the the needle geometry, during the simulation the needle is simply modelled using basic functions of the OpenGL Utility Library (GLU) and of the OpenGL Utility toolkit (glut); an opaque shiny gray cylinder is placed just below a green cube (see Figure 6.10(c)).

6.3 Clinical Validation

This section provides an attempt to answer the second part of our hypothesis by assessing the usefulness, the features and the performances of the simulator. Valid virtual reality simulators provide environments that closely mimic some characteristics of the environment in which the simulated task is performed. Medical VEs must be able to closely reproduce visual and real-time



(a)

(b)

(c)

FIG. 6.10: *Picture of a) a real medical co-axial Chiba needle, gauge 21, length 20 cm; b) a virtual needle created with* CosmoWorlds *VRML Editor; and c) the virtual needle used in the simulator.*

characteristics of the actual procedure. In interventional radiology, they must also provide realistic haptic feedback because IR procedures heavily relate to the sense of touch. The validation of medical simulators is critical to their eventual uptake in a training curriculum. However, the effectiveness and efficiency of IR VE training tools have not yet been addressed [44, 83]. The validation of commercial medical simulators should ideally include:

Content validation: estimation of what the VE tool trains, *e.g.* psychomotor skills or anatomy, *i.e.* does the simulation correctly replicate the steps of the target procedure? This is very important for use of a simulation in training or assessment. To achieve content validity the simulation should be based on a thorough cognitive task analysis.

Face validation: evaluation of resemblance between simulation and the procedure performed in the real world. It is generally achieved by comparing the opinions of experts with those of nonexperts.

Concurrent validation: assessment of the difference of performances between a group trained with the VE tool and a group trained with the traditional method. The assessment of both groups is held at the same point in time, so it is con-current.

Construct validation: assessment of the difference of performances between a group of experts with a group of nonexperts to check that the simulation correctly trains the skills required to perform the target procedure. So firstly the content must be right. Then there must be face validity to ascertain high fidelity of the simulator, so that relevant cues can be perceived. Then the operator's actions must produce correct, real world responses. The most critical actions should be detected, and therefore tested, by the simulation. So construct validity is likely to be present if the simulation can correctly differentiate experts from non-experts. Ideally, small differences in proficiency should be detectable.

Predictive validation: evaluation of the correlation between an assessment of an operator using the simulation, with an assessment of the same operator's performance in a patient at some later date, *i.e.* does performance in the simulation predict performance at some future time?

When assessing prototype VEs, content and face validations are initially most important for training purposes and these are the methods that we will use to assess our training system. For assessment we will also need construct validity. Then with known face, content and construct validity, it will be then appropriate to move on to seek evidence of skills transfer and predictive validation.

BIGN*e*PS*i* has been developed in close collaboration with practising interventional radiologists. In addition, we have made use of IR cognitive task analysis (CTA) studies that have been carried out by our collaborators [123]. An excerpt of freehand ultrasound-guided biopsy CTA is given in Table A.1. For feedback on each stage of development, informal face and content validation studies were conducted by medical collaborators to ensure that the fidelity of the simulator attains an acceptable level. Also, experienced IR practitioners from Europe and the US were allowed to experience the midterm version of the simulator in a large conference setting during a satellite workshop at the British Society of Interventional Radiologists (BSIR) 2005 conference. Many suggestions for improvements were collected and subsequently integrated into the simulator. For example, one issue identified was the placement of the haptic interfaces in relation to the virtual patient. Placing the Phantom devices side by side in front of the computer monitor is not an appropriate mapping to real world conditions. The Phantom devices both needed to be turned through 45 degrees to give a more natural interface and to avoid potential collisions between the haptic arms (see Figure 6.2), and the software was therefore updated to allow for this.

In further objective evaluations, questionnaires were developed to evaluate the usefulness, the features and the performances of the simulator. These were distributed to experts and inexperienced IR trainees at three different workshops. A mentor assisted trainees in the use of the simulator. These workshops were held at a large teaching hospital (Royal Preston Hospital, UK) at intervals of one year in July 2005, 2006, and 2007. The data obtained from these questionnaires and its analysis are shown in Sections 6.3.1, 6.3.2 and 6.3.3 respectively.

6.3.1 Early evaluation

Table C.1 shows the questionnaires used during the first workshop at the teaching hospital. This questionnaire was individually given to twenty inexperienced trainees. They were asked to fill out this questionnaire by circling the option that they most agree with to evaluate i) the visual rendering methods, ii) the haptic rendering methods, and iii) the usefulness of our virtual environment training platform. The resulting data is shown in Table 6.1. The data was obtained using the early version of the simulator that made use of an immersive virtual environment with stereoscopy, which enables the co-location of 3D graphical models with a single haptic joystick. This hardware platform includes a Phantom Desktop force feedback interface from SensAble

How much do you agree with the following statements		Average answer	Standard deviation
(1 = strongly disagree, 2 = moderately disagree, 3 = slightly disagree, 4 = don't know, 5 = slightly agree, 6 = moderately agree, 7 = strongly agree)			
(1)	The visual rendering is good enough for training	5.4	1.43
(2)	It is fundamental to hold two force-feedback devices at the same time	6.15	1.04
(3)	During the scanning of the body, the force feedback is realistic enough for a training tool	5.55	1.19
(4)	During the puncture, the force feedback is realistic enough for a training tool	4.7	1.59
(5)	When the simulator would be fully developed, you think it would be worth to initially train on the simulator	6.35	0.99

TAB. 6.1: *Data obtained from the questionnaires given to twenty inexperienced trainees during the first workshop at the teaching hospital, using the early version of the simulator described in Reference [265]. Trainees were asked to fill out this questionnaire by circling the option that they most agree with. The number of answers for each question is given as a percentage.*

Technologies [233], and a display monitor placed at a 45 degree angle that projects its image onto a semi-transparent mirror located above the force feedback device (see Figure 6.11). The effect for the user is that the 3D anatomy models appear to float in the volume of space below the mirror, and he/she can "touch" and interact with the models via the haptic interface. A 3D Space mouse can also be used to rotate the anatomical models [1]. During the simulation, a single haptic device was in use. It initially allowed the trainee to navigate through the CT dataset in order to find a suitable cross section image in which to guide a needle for the biopsy (see Figure 6.12(a)). The same haptic joystick was then used to simulate the virtual needle (see Figure 6.12(b)). The image computed during the simulation was presented to the operator on a second monitor as a 2D multiplanar reconstruction (MPR) image (*i.e.*, a resampled CT slice) or fluoroscopy image (see Figure 6.13). The needle insertion was simulated using a simple haptic model based on Hooke's law and the location of the needle within the simulated image was represented as a bright (CT) or dark (fluoroscopy) attenuation, the extent and shape of which conform to the computed needle location and angle through the baseline CT image data (see Figure 6.13(a)).

As Table 6.1 and Figure 6.14 illustrate, feedback on the usefulness of the early version of our virtual environment for training has been positive although one haptic device only was in use. However, it was unanimously accepted that two haptic devices should be simultaneously used, one for the ultrasound probe and another for the needle. Participants agreed that the visual rendering methods were at an acceptable level. The satisfaction feedback for the needle haptic

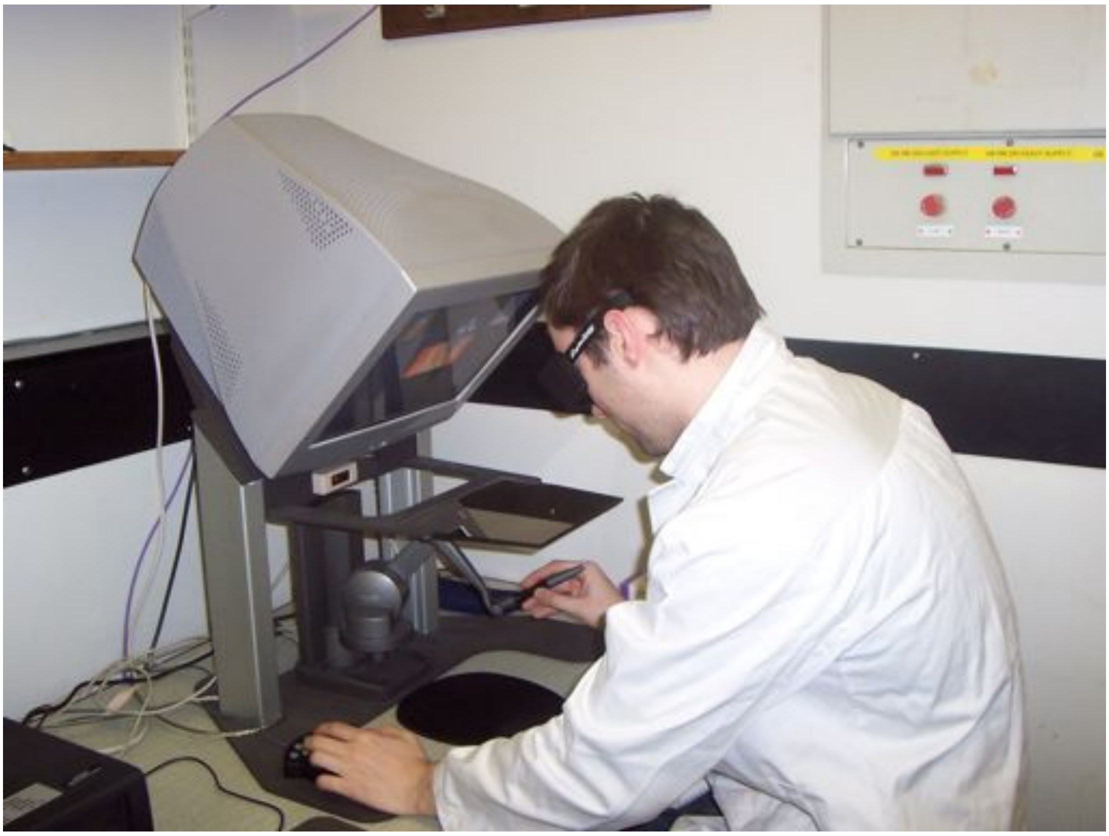
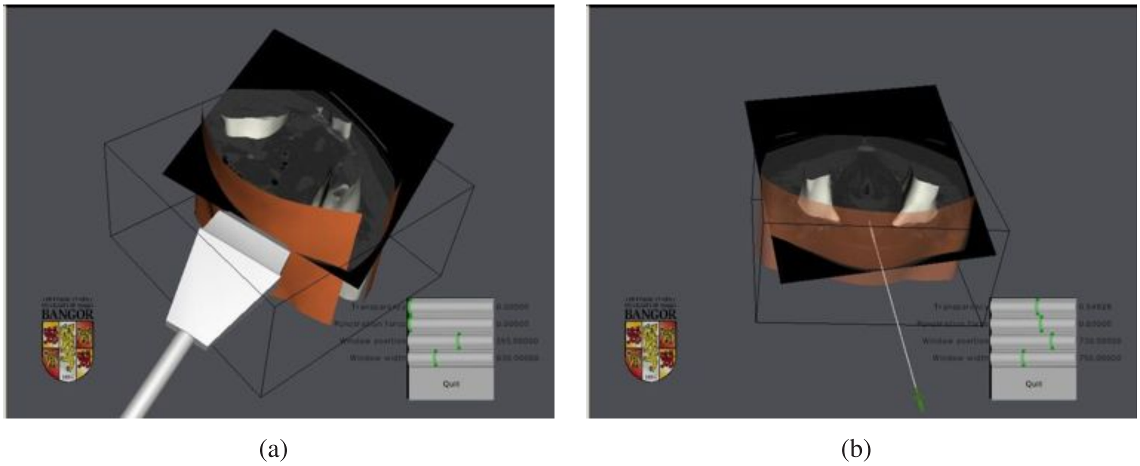


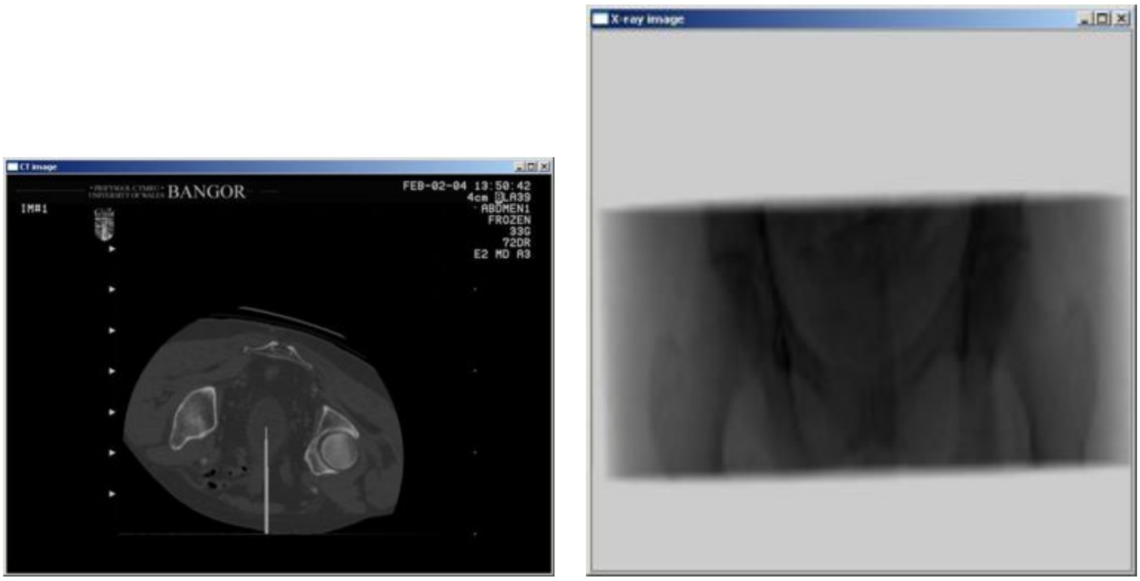
FIG. 6.11: *The early VE platform presented in [265]. It makes use of an immersive virtual environment with stereoscopy, which enables the co-location of 3D graphical models with a haptic joystick.*



(a)

(b)

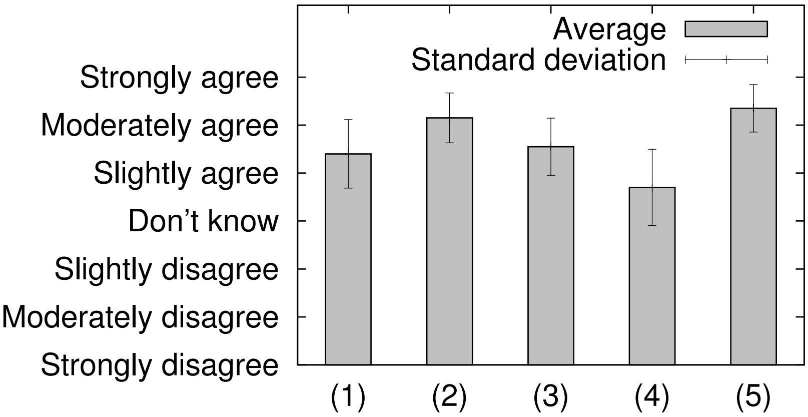
FIG. 6.12: *Simulation using the haptic interface as a: (a) probe; (b) needle. In this early version, the same haptic joystick is used to mimic the virtual ultrasound probe and the virtual needle.*



(a) 2D MPR of X-ray CT.

(b) X-ray-like image.

FIG. 6.13: *Guidance images used in the early simulator.*



(1) The visual rendering is good enough for training
(2) It is fundamental to hold two force-feedback devices at the same time
(one for the needle, one for the ultrasound probe)
(3) During the scanning of the body,the force feedback is realistic enough for a training tool
(4) During the puncture, the force feedback is realistic enough for a training tool
(5) When the simulator is fully developed,
it would be useful to initially train on the simulator

FIG. 6.14: *Validation of BIGNePSi's early version.*

rendering was lower and its standard deviation was higher than the satisfaction level for other features of the simulator. This could be due to the fact that only one type of internal tissue was taking into account during the puncture step, which was obviously not realistic. These results suggested that it was fundamental to improve the haptic model of the needle puncture. It also appeared that using stereoscopy was an useful feature to initially locate the puncture site. However, most of the trainees complained about the constraint of wearing goggles and removed them once the needle had penetrated the skin surface. Indeed, during the procedure on a real patient, once the needle is inserted, interventional radiologists only look at guidance images to steer the needle into the right target (see Figure 2.1).

6.3.2 Midterm evaluation

An identified issue with this early version of the simulator was related to the permanent use of stereoscopy glasses. The second workshop at the teaching hospital was held at half-term of this study. The simulator was used on a laptop without stereoscopy. Two haptic devices were attached to the computer and co-collocated with the virtual anatomy. The first acted like a virtual image probe, *e.g.* an ultrasound transducer. As the trainee moved the probe over the surface of the virtual body, he/she was provided with appropriate force feedback so that contact between the probe and the skin could be maintained. The image plane corresponding to the position and orientation of the probe was computed simultaneously and displayed in realtime as a sub-window of the virtual environment (see Figure 6.15). The second haptic device behaves like a needle and can puncture through the virtual skin and tissue layers. If the needle intersects with the image plane of the probe, then the trainee will see the needle on the reconstructed image.

Table C.2 shows the questionnaire that has been used. The data collected is given in Table 6.2 and Figure 6.16 is a summary. The participants were asked to evaluate i) the overall level of realism of the simulator (see Questions 7 and 8.h), ii) the rendering of guidance images (see Question 8.a), iii) the haptic feedback of the patient's skin on the US probe (see Question 8.b), iv) the haptic feedback of the needle puncture (see Questions 8.c and 8.d), and v) the usefulness of the simulator (see Questions 8.e, 8.f, and 8.g).

Participants

On the one hand, the simulator was exposed to nineteen trainees. Approximately all of them were inexperienced with their average duration spent in radiology practice being 1.68 years (see Ques-

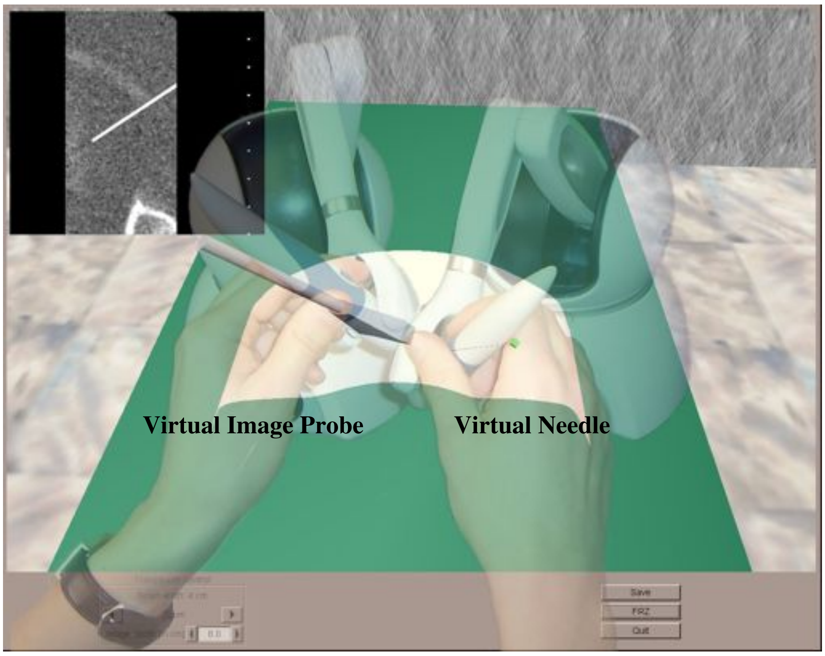


FIG. 6.15: *The midterm virtual environment platform using MPR of X-ray CT images for guidance, and two haptic devices (one to mimic the ultrasound probe, one to minic the needle).*

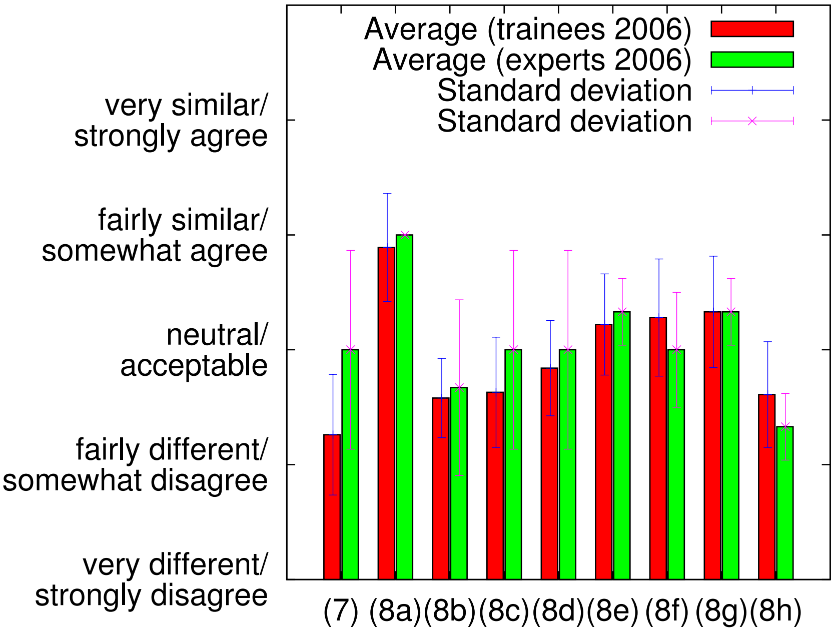


FIG. 6.16: *Validation of BIGNePSi's midterm version.*

Test subjects:		19 trainees		3 experts	
Questions		Average answer	Standard deviation	Average answer	Standard deviation
3.	Age:	30.42	2.59	42.33	11.85
4.	Number of years in radiology:	1.68	1.11	13.50	13.48
5.	How confident do you feel when performing imaging guided needle puncture procedures? (1=no confidence; 10=full confidence)	4.32	1.53	10.00	0
6a.	How many IR procedure performed this last 3 months (in total)?	8	6.97	N.A.	N.A.
6b.	independently?	2	4.26	N.A.	N.A.
7.	How similar did you find the procedure on the simulator model to the same kind of procedure in patient? (1=very different, 2=fairly different, 3=neutral, 4=fairly similar, 5=very similar)	2.26	1.05	3.00	1.73
8.	How much do you agree with the following statements: (1=strongly disagree, 2=somewhat disagree, 3=acceptable, 4=somewhat agree, 5=strongly agree) Simulation of the patient environment:				
a)	CT visual elements were of high quality	3.89	0.94	4.00	0
b)	Tactile (touch) elements were accurate	2.58	0.69	2.67	1.53
	Simulation of the needle insertion procedure:				
c)	Needle insertion through skin is realistic	2.63	0.96	3.00	1.73
d)	Needle insertion through tissues is realistic	2.84	0.83	3.00	1.73
	General:				
e)	The level of difficulty of the cases were appropriate	3.22	0.88	3.33	0.58
f)	The cases covered situations encountered in my practice	3.28	1.02	3.00	1
g)	The cases were a suitable challenge for my skills	3.33	0.97	3.33	0.58
h)	The simulator reflected a high level of realism	2.61	0.92	2.33	0.58

TAB. 6.2: *Data obtained from the questionnaires given to nineteen inexperienced trainees and three experts during the second workshop at the teaching hospital, using the version of the simulator shown in Figure 6.15, using two haptic devices (one with the needle puncture haptic rendering presented in Section 5.3.2, one with the proxy-based algorithm for the US probe), MPR of CT images for guidance, and running on a laptop without stereoscopy capability. Participants were asked to fill out this questionnaire by circling the option that they most agree with.*

tion 4), and only having performed a total of eight imaging guided needle puncture procedures in the previous three months (see Question 6.a). These eight procedures included two procedures performed independently (see Question 6.b). Note that answers at Questions 6.a and 6.b were relatively heterogeneous (standard deviations were 6.97 and 4.26 respectively). On the other hand, BIGN*e*PS*i* was also exposed to three experts. All three were fully confident in performing image guided needle puncture.

Overall Level of Realism of the Simulator

Whereas experts were neutral about how similar they found the procedure on the simulator model to the same kind of procedure in patients, trainees moderately considered that the procedure on the simulator model was dissimilar (2.26 for trainees and 3.00 for experts at Question 7). These results contrast to those at Question 8.h about the level of realism of the simulator. Indeed, trainees' level of satisfaction was enhanced and became superior to experts' satisfaction level (2.61 and 2.33 respectively). It was unanimously accepted that the level of realism was relatively poor due to the difficulty to correctly place the needle and the probe with respect to each other, particularly at the beginning of the puncture phase. To tackle this, the depth perception must be increased. It was suggested to i) enable the stereoscopy mode if possible, or ii) add the shadows projected by the needle and the probe onto the patient's body.

Rendering of Guidance Images

Both experts and trainees agreed that CT visual elements were of high quality (3.89 for trainees and 4.00 for experts). This feature was the least disputed in both groups, indeed results from Question 8.a got the lowest standard deviation.

Haptic Feedback of the Patient's Skin on the US Probe

On the other hand, both groups rated the tactile simulation of the patient below acceptable (2.58 for trainees and 2.67 for experts). It has been noticed that there was no tissue deformation when pushing on the patient's skin with the ultrasound probe. It has also been noted that internal structures such as bony landmarks were not taken into account. This was due to the fact that the haptic rendering of the skin surface was performed using the proxy-based algorithm.

Haptic Feedback of the Needle Puncture

Once again experts rated higher this feature. Indeed experts were neutral about the level of realism of the haptic rendering of needle puncture through the skin and through tissues, whereas trainees slightly disagreed. This feature was not particularly disputed amongst trainees (standard deviations were 0.96 and 0.83 for Questions 8.c and 8.d respectively). This contrasted with answers given by experts, for both questions the results collected ranged from 1 to 4 and standard deviation was 1.73.

Usefulness of the Simulator

In general, both novices and experts gave a positive feedback on the appropriateness of the cases that BIGN*e*PS*i* provided for training.

General Observations

One of the main issues identified in the early version of the simulator was the compulsory use of stereoscopy goggles during the whole procedure. This was made optional in the midterm version of BIGN*e*PS*i* as stereo is not needed for the second stage of the needle puncture procedure, and the requirement to wear stereo glasses can be removed. Therefore, during the second workshop at the teaching hospital, the simulator was exposed to experts and inexperienced trainees using an off-the-shelf laptop without stereoscopy. However, this made much more difficult to place the virtual probe and needle at the skin entry site. This can be tackle by using polarised glasses with planar displays, or auto-stereoscopic displays, and it will not have the drawbacks of conventional stereoscopic displays that make use of CRT monitors. As part of the general comments about the simulator, a trainee - who has spent two years in radiology and rated his/her level of confidence as six - reported:

> “Good simulation overall, improved from last year. Much more improvements possible.”

6.3.3 Final evaluation

To further improve the perception of depth, in the final version of BIGN*e*PS*i*, we added shadowing of the needle and ultrasound probe on the patient’ skin (see Figure 6.2). However, when the needle is being inserted through the tissues, the visual cue is exclusively taken from the 2D guidance image being displayed and the user can interactively enable or disable the stereoscopy.

Table C.3 shows the questionnaire that has been used at the third workshop at the teaching hospital. The data collected is given in Table 6.3 and Figure 6.17 is a summary. As above, the participants were asked to evaluate i) the overall level of realism of the simulator, ii) the rendering of guidance images, iii) the haptic feedback of the patient's skin on the US probe, iv) the haptic feedback of the needle puncture, and v) the usefulness of the simulator.

Test subjects:		10 trainees		5 experts	
Questions		Average answer	Standard deviation	Average answer	Standard deviation
3.	Age:	30.11	3.1	43.40	3.58
4.	Number of years in radiology:	1.0	0.0	15.60	5.5
5.	How confident do you feel when performing imaging guided needle puncture procedures? (1=no confidence; 10=full confidence)	3.50	2.06	8.70	0.97
6a.	How many IR procedure performed this last 3 months (in total)?	6.94	8.72	N.A.	N.A.
6b.	independently?	0.0	0.0	N.A.	N.A.
7.	How similar did you find the procedure on the simulator model to the same kind of procedure in patient? (1=very different, 2=fairly different, 3=neutral, 4=fairly similar, 5=very similar)	2.29	0.95	2.80	1.1
8.	How much do you agree with the following statements: (1=strongly disagree, 2=somewhat disagree, 3=acceptable, 4=somewhat agree, 5=strongly agree)				
	Simulation of the patient environment:				
a)	US visual elements were of high quality	3.56	0.53	3.80	0.45
b)	Tactile (touch) elements were accurate	2.44	0.88	3.00	1.41
	Simulation of the needle insertion procedure:				
c)	Needle insertion through skin is realistic	2.44	0.53	3.00	1.41
d)	Needle insertion through tissues is realistic	2.67	0.87	3.25	1.5
	General:				
e)	The level of difficulty of the cases were appropriate	3.33	0.87	4.00	1.41
f)	The cases covered situations encountered in my practice	3.56	0.73	N.A.	N.A.
g)	The cases were a suitable challenge for my skills	3.67	0.87	N.A.	N.A.
h)	The simulator reflected a high level of realism	2.44	0.73	3.67	0.58

TAB. 6.3: *Data obtained from the questionnaires given to ten inexperienced trainees and five experts during the third workshop at the teaching hospital, using the version of the simulator shown in Figure 6.2, using two haptic devices (one with the needle puncture haptic rendering presented in Section 5.3.2, one with the US probe haptic rendering presented in Section 5.3.1), US simulated images for guidance, and running on a PC with stereoscopy capability. Participants were asked to fill out this questionnaire by circling the option that they most agree with.*

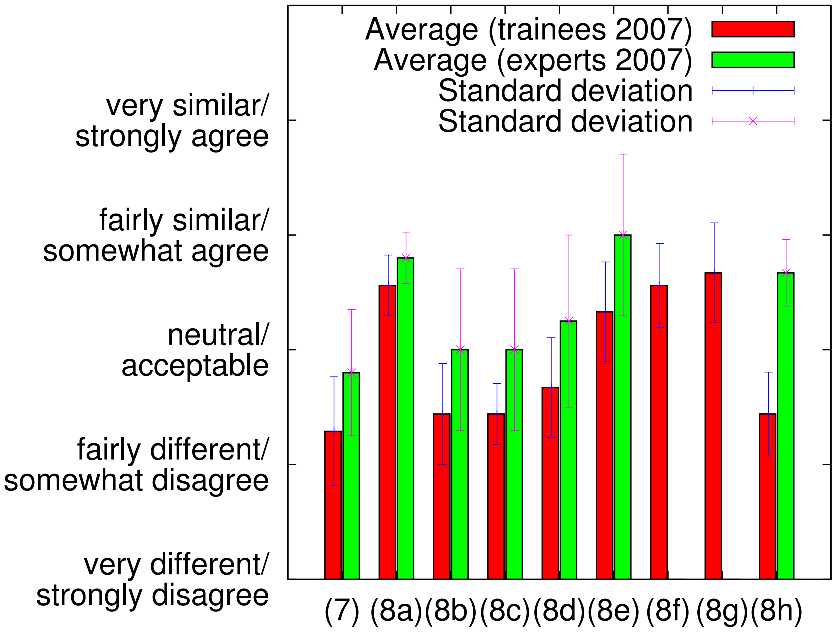


FIG. 6.17: *Validation of BIGNePSi's final version.*

Participants

The simulator was exposed to ten inexperienced trainees who had spent only one year in radiology. They performed a total of seven imaging guided needle puncture procedures during the last three months. None of these procedures were performed independently. BIGN*e*PS*i* was also exposed to five experts fully confident in performing image guided needle puncture and have averagely spent 16 years in radiology.

Overall Level of Realism of the Simulator

Whereas experts were neutral about how similar they found the procedure on the simulator model to the same kind of procedure in patients, trainees moderately considered that the procedure on the simulator model was not similar (2.29 for trainees and 2.8 for experts at Question 7). The results were relatively similar to the results at the same question in the evaluation of the mid-term version of BIGN*e*PS*i*. On the other hand, the results for Question 8.h about the level of realism of the simulator slightly decreased for trainees and substantially increased for experts (2.44 and 3.67 respectively). Following recommendations made during the evaluation of the mid-term version, increasing the depth perception using shadowing and stereoscopy helped to improve the level of fidelity of the simulator.

Rendering of Guidance Images

Both experts and trainees agreed that US visual elements were of high quality (3.56 for trainees and 3.80 for experts). Once again, this feature was the least disputed in both groups, indeed results at Question 8.a got the lowest standard deviation.

Haptic Feedback of the Patient's Skin on the US Probe

Participants from both groups noticed that there was no tissue deformation when pushing on the patient's skin with the ultrasound probe. However, enabling the possibility to haptically render internal hard structures such as bony landmarks helped improving the tactile simulation of the patient. Indeed, feedback on the tactile simulation of the patient increased for the group of experts (from 2.67 to 3.00). Nevertheless, results from this question slightly decreased for the group of trainees (was 2.58, now 2.44).

Haptic Feedback of the Needle Puncture

Once again experts rated higher this feature. The level of satisfaction of experts for the haptic feedback of the needle puncture slightly improved since the mid-term evaluation. Although this feature is still disputed amongst experts, the standard deviations decreased. This feature was less disputed amongst trainees.

Usefulness of the Simulator

Feedback about usefulness of the simulator slightly improved. In general, both novices and experts gave a positive feedback on the appropriateness of cases that BIGN*e*PS*i* provided for training.

General Trend

Figure 6.18 combines results of the evaluation of both the midterm and the final versions. It shows three general trends:

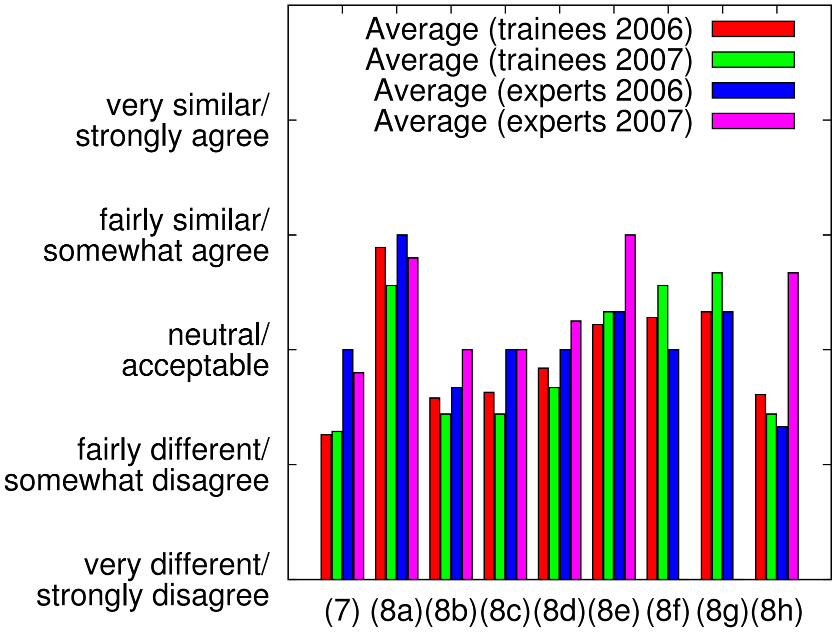


FIG. 6.18: *Validation of BIGNePSi's midterm and final versions.*

- in general, technical features of the simulator have been improved through the integration of the enhancements suggested during our preliminary demonstrations at workshops amongst IR students or practitioners.

- trainees' answers were relatively homogeneous whereas experts' answers were sometimes more disparate (see standard deviations in Figures 6.16 and 6.17). Although experts often gave higher scores than trainees, we observed during these workshops that in general experts were more careful about features and details of the simulator than novices. We conclude that experts had a higher expectation of what was required from BIGN*e*PS*i* based on their own experience whereas trainees were using BIGN*e*PS*i* as a training tool only.

- experts' satisfaction levels were generally higher than trainees' satisfaction levels about i) the similarity of the procedure on the simulator model to the same kind of procedure in a real patient, ii) the fidelity of the simulator, iii) the visual rendering and iv) haptic rendering. In contrast, trainees rated higher the level of complexity of simulated cases than experts. These might be explained by a bias due to the relative inexperience of trainees. However, to confirm this hypothesis, more research needs to be carried out. This includes a concurrent validity test where novice and expert performance will be compared and this is currently being planned. Indeed, BIGN*e*PS*i* has shown satisfactory performance in terms of haptic force feedback, system ergonomics, image clarity, and the usefulness of the simulator. Therefore the simulator has reached a point where it is now suitable for a concurrent validity test and BIGN*e*PS*i* has been instrumental in the success of a grant award to the Department of Health that will aim to develop an ultrasound-guided needle puncture of the viscera that is clinically validated. The quote as follows provide feedback from experts:

- A consultant who has spent 12 years in radiology:

  > "Huge potential!"

- A consultant who has spent 13 years in radiology:

  > "Excellent project. Probably more applications available than time to do them."

- A consultant who has spent 20 years in radiology:

  > "Simulated US image is good in many respects..."

- A consultant who has spent 25 years in radiology:

  > "Haptics good!"

6.3.4 Limitations

The evaluation of medical simulators developed in a context similar to BIGN*e*PS*i* generally lack of clinical validation to assess their actual usefulness, features and performances. In contrast, this section provided an attempt to evaluate these in BIGN*e*PS*i*. However, this initial evaluation is at its early stage and suffers from some limitations. In particular, statistically the number of trials is relatively low to be fully significant and the results may be misleading. This is due to the difficulty to regroup a sufficient number of trainees and experts to test the simulator at each development stage. Moreover, to be fully convincing, exactly the same group of trainees and experts would need to take part to the experiment at each development stage. However, this is not realistically achievable for the same reason as above. More research needs to be conducted to carry out an independent evaluation of the simulation. BIGN*e*PS*i* has been instrumental in the success of a grant award to the Department of Health. This project is addressing the above limitations.

6.4 System Testing

6.4.1 BIGNePSi Testing Platforms

We have produced a configurable and validated training tool to enable IR practitioners to learn how to carry out image-guided needle puncture. A goal has been to provide a cost effective solution. Inexpensive components are therefore used wherever possible, and the configuration used can be tailored to match a variety of budgets, e.g. stereoscopy will be activated when a professional graphics card with quad-buffer is detected.

Table 6.4 is a summary of the current cost of haptic devices that are supported by the simulator. Table 6.5 shows the configuration, the cost and the performance of the platforms which have been used to test our training simulator. This includes: a) a high-end graphics workstation with a professional graphics card running Microsoft Windows XP, b) a standard PC with a recent gaming graphics card running Microsoft Windows XP, c) the same PC running Linux OpenSuse 10.2, d) a low-end PC with an "out-dated" graphics card, e) a laptop. Performances are given in number of frames per second (FPS). BIGN*e*PS*i* has been tested on a variety of platforms; from a laptop to a standard desktop PC with an ATI Radeon X1950 PRO to a state-of-the-art workstation with the far more powerful NVIDIA Quadro FX 3500 graphics card; running Microsoft Windows XP or Linux OpenSuse 10.2. These tests proved that the training system is portable and provides interactive performance in all cases.

Device	Cost (in £)
SensAble Omni	$\approx$ £$1,200$
Mantis Mimic	$\approx$ £$5,000$
SensAble Desktop	$\approx$ £$8,000$
SensAble Premium 1.5	$\approx$ £$20,000$

TAB. 6.4: *Cost of haptic devices that are supported by our training simulator.*

Type	Processor(s)	RAM (in GB)	Graphics cards	Cost	Performance (in FPS)
PC	Dual Intel Xeon 2.8 GHz	2	Nvidia Quadro FX 1000	N.A.	≈ 11
PC	Intel Core 2 Duo E6600 (2.4 Ghz)	2	ATI Radeon 1950 PRO	$\approx$ £600	≈ 15
Laptop	Intel T2400 (1.83 GHz)	2	Nvidia Quadro FX 350M	$\approx$ £$1,000$	≈ 30
PC	Intel Core 2 Duo E6600 (2.4 Ghz)	2	Nvidia Quadro FX 3500	$\approx$ £$1,300$	≈ 43

TAB. 6.5: *Cost and performance of some platforms used to test our training simulator.*

6.4.2 Physics-based X-ray Simulation Testing Platforms

In Section 4.7, we proposed a method to simulate x-ray images on the GPU. Two versions of the algorithm have been produced:

CPU-based. The algorithm is well suited for parallel programming. Indeed, each processor available can compute simultaneously a different sub-region of the final image. Thus the algorithm has been implemented using the pthread library to take advantage of parallel computing on multi-processor computers (i.e. with several physical CPUs) or mono-processor computers with hyper-threading (i.e. with only one physical CPU but two logical CPUs).

GPU-based. We take advantage of 3D textures and of the parallel processing capabilities of this programmable hardware. The fragment program converts texture values into attenuation coefficients and compute the x-ray attenuation by ray-tracing.

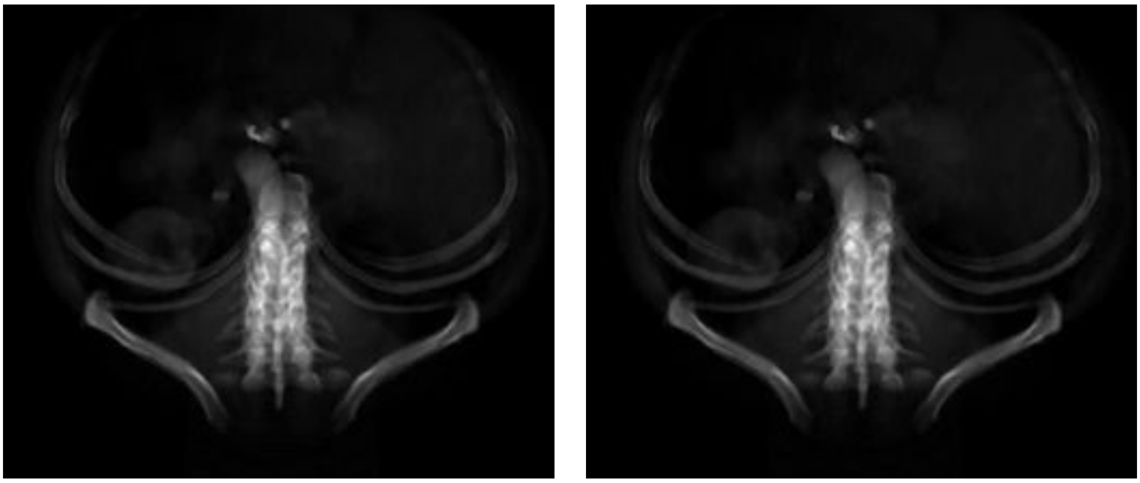
Table 6.6 summarises the performance achieved on different hardware configurations. The time taken (in seconds) to perform a predefined animation of 1000 512×512 frames generated from a $512 \times 512 \times 62$ CT data set is recorded. Figure 6.19 shows examples of the computed images.

	Bi-Xeon 2.4 GHz without hyper-threading	Bi-Xeon 2.4 GHz with hyper-threading	Nvidia GeForce 6400	Nvidia Quadro FX 3400
CPU x-ray attenuation	< 1	< 1	N.A.	N.A.
GPU x-ray attenuation	N.A.	N.A.	≈ 500	≈ 1000

TAB. 6.6: *Performance comparison: average number of* 512×512 *images computed in one second using a* $512 \times 512 \times 62$ *CT data set.*

The parallised CPU-based simulation of high-resolution images is too slow on bi-processors and prevents real-time interaction. Hyper-threading seems not to improve the performance. However, our GPU-based implementation enables the simulation of x-ray images from volume data in real time, at high resolution, and with the same image quality as software volume rendering on CPUs. This difference in term of performances can be explained as follows:

- the GPU architecture is highly parallel,
- the access from the main memory to fetch the attenuation coefficient of voxels is slower than fetching texel values from the 3D texture,



(a) Image computed on the CPU.

(b) Image computed on a laptop with a Nvidia GeForce 6400 graphics card.

FIG. 6.19: *Examples of x-ray simulations.*

- the implementation of mathematical operations such as the exponential function is more efficient on GPUs.

The use of the attenuation law also results in more accurate x-ray images than previously reported work on GPUs.

Chapter 7

CONCLUSIONS AND FUTURE WORK

Contents

7.1 Contributions

At the beginning of this thesis, we hypothesised that a flexible and cost effective integrated visual-haptic system for the simulation of image guided needle puncture, can be realized using off-the-shelf components (visual displays, haptic devices and work stations), to deliver a high fidelity training experience. The Bangor Image Guided NEedle Puncture SImulator (BIGN*e*PS*i*) project has been an investigation of this hypothesis.

Chapter 3 presented an overview of principles and applications of computer graphics in medicine. We noted that medical simulators have a significant role to play in providing a viable alternative to traditional apprenticeships in patients, particularly for the training of core skills. However, the worldwide market for such simulators is still relatively small, and only a few companies are offering commercial products today. One factor has been the high cost associated with purchasing and maintaining the state-of-the-art equipment required. In this thesis, we have presented novel techniques that have been integrated to build one of the first simulators to provide interventional radiology (IR) training for the commonly performed procedure of image guided needle puncture.

The work carried out through this research and its results demonstrated the feasibility of such a simulator. Chapters 4 and 5 presented the main contributions of this work for the visual and haptic components of the simulator respectively. This includes:

Needle Tracking in 3D Textures For the rendering of ultrasound or x-ray images, full advantage has been taken of 3D texture hardware. However, in general it is not possible to update the content of the 3D texture in realtime to change the voxel intensities at the needle position because of the overhead involved in transferring the recalculated data from main memory to the graphics card. In Section 4.4, we described a new method to update the 3D texture corresponding to a patient's CT volume dataset to track the needle in realtime during the puncture. It makes use of OpenGL ® Shading Language (GLSL) to avoid the limitations of the fixed graphics rendering pipeline and to take advantage of programmable graphics hardware. The principle is to detect if the texture coordinates corresponding to each pixel of the image to render, computed from the 3D texture, are being penetrated by the needle.

Adding Lesions in 3D Textures In Section 4.5, we demonstrated that simulated tissue lesions can be added at any position within the patient dataset using a similar technique to the above.

Interactive Ultrasound Simulation from CT Data Simulating ultrasound images in realtime was a challenging task because of the conflicting demands of interactivity and accuracy. Although computer hardware gets continuously faster, physically-based simulation still remains a demanding task in terms of computational time. Tissues correspond to small non-uniform structures. When an ultrasound wave hits such an irregular surface, this obstacle behaves as a secondary sound source and it emits a wave in many directions. Such an approach was therefore not suitable for this use in a real-time virtual environment. We demonstrated in Section 4.6 that a possible alternative approach was to approximate ultrasound images by adding, to the original CT data, artefacts characteristic of US images. The graphics pipeline for rendering US images from a CT volume takes full advantage of modern graphics processing unit (GPU) abilities. The algorithm is based on a multi-pass algorithm implemented using framebuffer object (FBO) and GLSL when applicable, combined with processing on the CPU when no GPU alternatives are possible.

Interactive Physically-based Simulation of X-ray Images from CT Data Fluoroscopic (*i.e.* x-ray) images are sometimes used to guide needles, and this is the image modality that is commonly use to guide catheters and guidewires. Simulating such images using the physical laws has been extensively studied in physics. Real-time performance can be almost achieved using polygon meshes when only the attenuation of x-rays is computed by ray tracing. However, the physically-based simulation of x-ray images at interactive framerate from volume data sets still remained a challenge. We demonstrated that the attenuation law can be also applied in realtime on the GPU to simulate x-ray images from volume datasets. It makes use of an effective implementation, written in GLSL, of ray-tracing through voxel data to compute the attenuation of x-rays with regard to the length of the ray path across each voxel crossed by the incident beam.

Ultrasound Transducer Haptic Model The classic proxy-based algorithm is commonly accepted for the haptic rendering of surfaces. In this model, the haptic device interacts with the surface of a 3D object represented as a triangular polygon mesh. Obviously, the main drawback of such an approach is that there is no information about the content of internal structures within the polygon mesh, *i.e.* this model does not natively take into account internal hard structures such as bony landmarks. In Section 5.3.1, we demonstrated how the proxy-based haptic rendering method can be extended to use volumetric data so that the trainee can feel underlying structures, such as ribs, whilst scanning the surface of the virtual patient's body with the ultrasound transducer. The method is based on ray-tracing through the voxel data to detect if any hard structure is located between the proxy and the actual device position. When this is the case, the proxy position is located at the entry into the hard structure, and the stiffness is increased. This method does not introduce any instability in the haptic rendering because the force returned to the user is always parallel to the force normally computed with the classic proxy-based algorithm.

Needle Puncture Haptic Model Our clinical collaborators have been collecting *in vitro* data of needle insertion forces in animal porcine and ox tissues. However, there is still a shortage of experimental data on tissue resistance. To date, only resistance data of the kidney and the liver is available for punctures with different kinds of needles. In this research, only data for Chiba needles (21 and 22 gauge) has been of interest as this is the needle that is most often used for gaining access to the kidney, which is a typical example of the visceral interventional needle puncture procedures that we were simulating. In Section 5.3.2, we estimated analytic models of Chiba needle insertion forces in the liver and the kidney. However, due to the shortage of resistance data for other kinds of tissue, these models cannot be directly implemented into our haptic model of needle puncture. Therefore, in Section 5.3.2, we described a flexible volume haptic model of Chiba needle puncture that aimed to reproduce features observed in the experimental data, *i.e.* an initial displacement before the penetration of the tissue, then a rebound into the force, then a plateau phase.

The benefits of the different contributions summarized above were illustrated by their use in an IR training simulator for ultrasound guided needle puncture, where a trainee interacts with a virtual patient using two haptic devices, one for the US probe, one for the needle. Section 6.2 described how the simulator has been built using these components.

The validation of the simulator is however critical to its eventual uptake in a training curriculum and a project such as this could not be undertaken without close co-operation with the domain experts. Section 6.3 detailed the clinical validation results obtained as well as technical results. For feedback on each stage of development, face and content validation studies were conducted by medical collaborators to ensure that the fidelity of the simulator attained an acceptable level. In further, objective evaluations, questionnaires were developed to evaluate the features and the performances of the simulator. These were distributed to trainees and experts at three different workshops. These workshops were at a large teaching hospital, and were held at intervals of one year. At both of these the simulator was exposed to inexperienced IR trainees and experts, and the data were obtained from the questionnaires. At a fourth workshop, experienced IR practitioners from Europe and the US were also allowed to experience the simulator in a large conference setting. Many suggestions for improvements were collected during these workshops and subsequently integrated into the simulator. Through such enhancements, the simulator has reached a point where it is now suitable for a concurrent validity test (where novice and expert performance will be compared) and this is currently being planned. Indeed, our preliminary demonstrations during workshops amongst IR students or practitioners have demonstrated satisfactory performance in terms of haptic force feedback, system ergonomics, image clarity, and the usefulness of the simulator.

7.2 Discussion and Future Work

The previous section provided evidence that our hypothesis has been proved and that it is possible to use off-the-shelf components to build a flexible and cost effective integrated visual-haptic system for the simulation of image guided needle puncture. However, although the simulator is ready for clinical validation, there are still several limitations that can be identified and functionalities that can be added:

- First, there is no tissue deformation, such as between the ultrasound transducer and the patient's skin surface, which could improve the depth cue to help the trainee to co-locate the two haptic devices with respect to the virtual patient. Keeping in mind the conflicting demands of interactivity and accuracy, current state-of-the-art for in fast soft-tissue modelling, such as 3D ChainMail methods [78, 226, 152], mass-spring techniques or finite element models [145, 135], would provide efficient solution to tackle the lack of deformations between the image probe and the patient's skin.

- Also, the lack of tissue motion and deformation when the needle is penetrating tissues has been identified as one of the main limitations of the simulator. Indeed, the movement of tissues is often used by interventional radiologists to assess the position of the needle into the real-time US images. For example, when the needle is not directly visible in the ultrasound image, some interventional radiologists may shake the needle to identify where tissues are moving and deduce the position of the needle. Therefore, to allow the trainee to locate and track the needle during the simulation, the needle appears clearer using our virtual ultrasound scanner than it will with an actual ultrasound procedure.

- The real-time physically-based simulation of ultrasound waves still remains a challenge due to the computational time required. Therefore, the simulation of ultrasonic images from CT data presented in this thesis is purely a visual simulation, by contrast with physically-based simulations [108]. The use of GPGPU techniques, in particular CUDA, to address this still remains to be investigated. The technique presented in this thesis is based on adding characteristic US artefacts to the CT data. Therefore, the validation is an assessment of the virtual images by practitioners. Intensities in the simulated US image are computed using a LUT. When contrast agent is injected in blood vessels, their intensities in CT data become similar to those of bones. Consequently, blood vessels may generate false shadowing in the simulated US images.

- Due to the shortage of experimental data on tissue resistance, estimated analytic models of Chiba needle insertion forces in the liver and the kidney could not be directly used. Then, the proposed haptic model for needle insertion is also based on a LUT and it mimics features observed in experimental data. Resistance force for more kinds of tissue need to be studied. This will enable to replace the coefficients extracted from this haptic LUT by a coefficient computed from analytic models that accurately match the experimental data.

- Also, selected off-the-shelf haptic devices cannot output a maximum constant force that matches measurements. Support for larger forces with devices that apply force directly to the point of contact through tension in cables needs to be investigated.

- The haptic device used to mimic the needle could be mounted on the clinical force study's experimental bench to record simulated resistance forces. These will be compared with force resistance data of actual tissues in order to finely tune stiffness coefficients, and finally to validate that the haptic model can reproduce realistic similar forces.

- One issue identified was the placement of the haptic interfaces in relation to each other. Indeed, when the virtual needle is next to the US probe, the mechanical arm of devices may collide between each other.

- Warmth to touch, movement both of the patient and of internal structures (bowel and respiratory movement and pulsation of vessels) will also need to be studied [268].

- Audio feedback to the trainee as a result of pain or discomfort could also be studied.

- Comprehensive construct and concurrent validation, and skills transfer studies, form a key part of future, proposed work. This should be conducted within IR curricula to evaluate the clinical effectiveness of the simulator to train visceral needle puncture procedures.

- The prototype of the algorithm has been tested on platforms with different graphics cards. It successfully runs on computers equipped with a relatively recent GPU, such as Nvidia GeForce 6400 on laptops and Nvidia Quadro FX 3400 on professional workstations. Most graphics cards of the previous generation, such as the Nvidia Quadro FX 1000 graphics card used in our development workstation, does not support a “while loop” necessary for the implementation of the algorithm. Consequently, the algorithm would be executed in such a case without any hardware acceleration. Thus the algorithm could not be integrated and tested into the final version of our image guided needle puncture simulator and this remains to be done.

Despite these limitations, BIGN*e*PS*i* is proving to be a valuable achievement of this research. In particular, it has been instrumental in the success of two grant awards, one to the Engineering and Physical Sciences Research Council (EPSRC), and one to the Department of Health. These two projects are addressing the above limitations.

Appendix A

EXCERPT OF FREEHAND ULTRASOUND-GUIDED BIOPSY CTA

Cognitive task analysis (CTA) of freehand ultrasound-guided biopsy aims to identify, describe and detail the whole process of this medical procedure, from the cues perceived, to the cognitive thought processes of decision-making, to the ensuing physical actions. This study has been performed by our psychologist partners of the Collaborators in Radiological Interventional Virtual Environments (CR*a*IVE) consortium [41, 123]. This work has been used extensively to guide the design of the simulator described in this thesis. An excerpt from this detailed task analysis is given in Table A.1.

Steps of task	
1.	Place probe on patient at incision site
2.	Looking at the ultrasound screen roll the probe back and forth on the incision site until the planned biopsy site is identified
3.	Note the depth of the planned biopsy site using the markers on the ultrasound screen
4.	Pick up the biopsy trocar (with needle) from the green sheet - make note of the measurement markers on the trocar to identify how deep this needle is going to be inserted
5.	Holding the needle shaft position the biopsy trocar at the incision site
6.	Holding the shaft begin to insert the trocar perpendicular to the skin into incision
7.	Position the trocar, probe and planned biopsy site in the same plane (to ensure all necessary information can be viewed on the ultrasound screen)
8.	Looking at the ultrasound screen jiggle the trocar gently and identify the trocar on the image (a moving needle is easier to identify than a static needle) and the planned biopsy site (the aim is to be able to see both the trocar and the planned biopsy site in the same image)
9.	Slowly advance the trocar towards the planned biopsy site, constantly viewing the ultrasound image to guide orientation and progress
10.	Continue advancing the trocar in this way until the planned biopsy site is reached (the needle should be positioned just short of the planned sample site, if a tumour is being biopsied the needle needs to be at the edge of the tumour), be guided both by the image and by your knowledge of the depth of the planned biopsy site
11.	Request patient to hold their breath
12.	Whilst patient holding their breath check (on the ultrasound screen) positioning of trocar is correct. Is the trocar in the correct position for biopsy? • Yes – go to next step (allow patient to breath, shallowly) • No – repeat from step 7 (if necessary withdraw the trocar slightly before proceeding) (allow patient to breath)

TAB. A.1: *Excerpt from [122] related to ultrasound guided needle puncture using a coaxial biopsy needle. The detailed description of each step of the task has been essential information for designing the simulator described in this paper.*

Appendix B

VISUAL RENDERING USING HARDWARE ACCELERATION

B.1 Volume Data Pre-processing

The input volume data is pre-processed using ITK as follows:

Extracting 3D Data from DICOM Files Medical scanners usually store each slice of a 3D dataset into a single DICOM file. Generally, files of a given dataset are located in the same directory, a directory containing one dataset only. Consequently, the first step is to extract each slice of a dataset, and store all of them into a single file. For efficiency and simplicity, we use a MetaImage header text file for raw images as specified in Insight Segmentation and Registration Toolkit (ITK) [134]. This is a fairly simple and comprehensible format consisting of a text header and a binary data section. The filename extension of MetaImage header text files is either ".mha" or ".mhd". In our simulator, the binary data section is written in a separate raw image file. Table B.1 is an example of MetaImage header, it is associated with the file *pa1_st1_se5_original_data.raw*.

Thresholding Medical CT scanners usually use the $[-1000, 1000]$ scale. Grey lines in Figure B.1 show paths of x-rays at different angles. Due to the data acquisition process and to the tomographic reconstruction principles, values are only known at the centre of a slice, which is the only common area crossed by the x-ray beam at each projection angle (see non-black area in Figure B.1). When many projections are acquired at different angles, this area describes a circle. Values in the black area are therefore unknown. However, image file formats are not generally designed to store circular images. Discrete image data is usually stored as a rectangular matrix of pixels. Scanner manufacturers usually set an arbitrary

```
ObjectType = Image
NDims = 3
BinaryData = True
BinaryDataByteOrderMSB = False
TransformMatrix = 1 0 0 0 1 0 0 0 1
Offset = -196.2 -190 -15
CenterOfRotation = 0 0 0
AnatomicalOrientation = RPI
ElementSpacing = 0.742188 0.742188 0.625
DimSize = 512 512 370
ElementType = MET_SHORT
ElementDataFile = pa1_st1_se5_original_data.raw
```

TAB. B.1: *Example of image file metaheader as specified in ITK [134]. The filename extension of MetaImage header text files is ".mha" or ".mhd"*

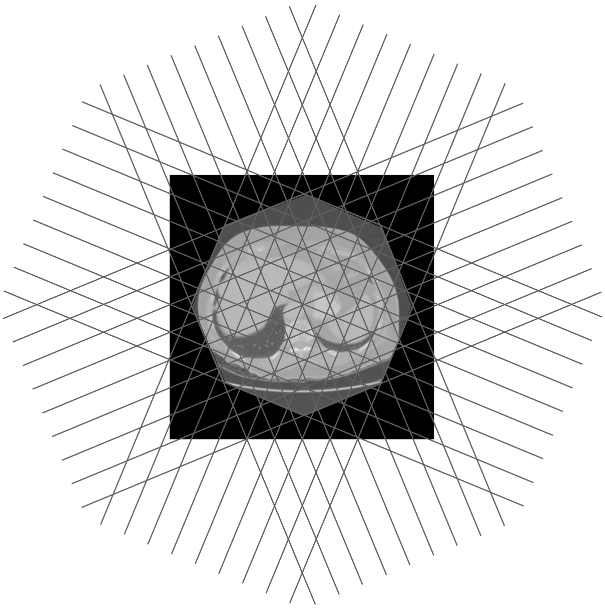


FIG. B.1: *CT slice acquisition and pixel padding – grey lines correspond to the X-ray beam at different angles – pixel values are known into the centre of the CT slice, which is the only common area crossed by the X-ray beam at each projection angle during the acquisition process – in black, pixel values correspond to pixel padding, a constant outside the range* $[-1000, 1000]$.

negative constant value, called *pixel padding value*, that is not in the $[-1000, 1000]$ scale (scanner manufacturers may use different values for pixel padding). Care must be given to remove such “invalid” values in the dataset (see Figure B.2). Using a simple threshold filter, values which are below -1000 are replaced by -1000, the Hounsfield value of air. Similarly, values above 1000 are replace by 1000, the value of bones.

```
Data: DICOM data set in Hounsfield unit (HU)
Result: texture element (texel) values between -1000 and 1000
begin
    /* To remove pixel padding, clamp each Hounsfield value ($H$) of the
       original DICOM data set between -1000 and 1000                    */
    foreach voxel $H$ of the dataset do
        /* Clamp $H$ between -1000 and 1000                              */
        if $H < -1000$ then $H \leftarrow -1000$
        else if $H > 1000$ then $H \leftarrow 1000$
    endfch
end
```

FIG. B.2: *Remove pixel padding in original DICOM data*

Low-pass Filtering Image data acquired from any kind of scanner will be degraded and will inevitably contain noise. Figure B.3 shows how the image degradation process and the image restoration process are usually modelled. In the 2D spacial domain, the degradation process can be written as follows:

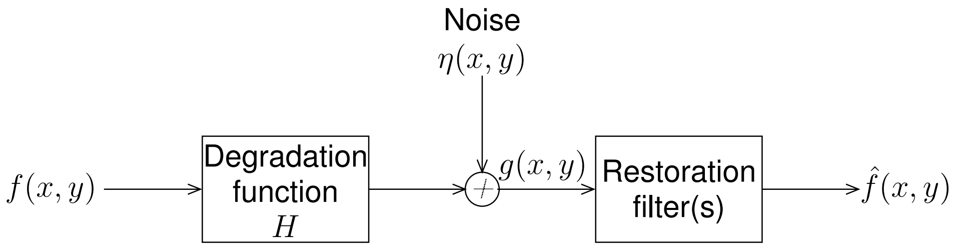
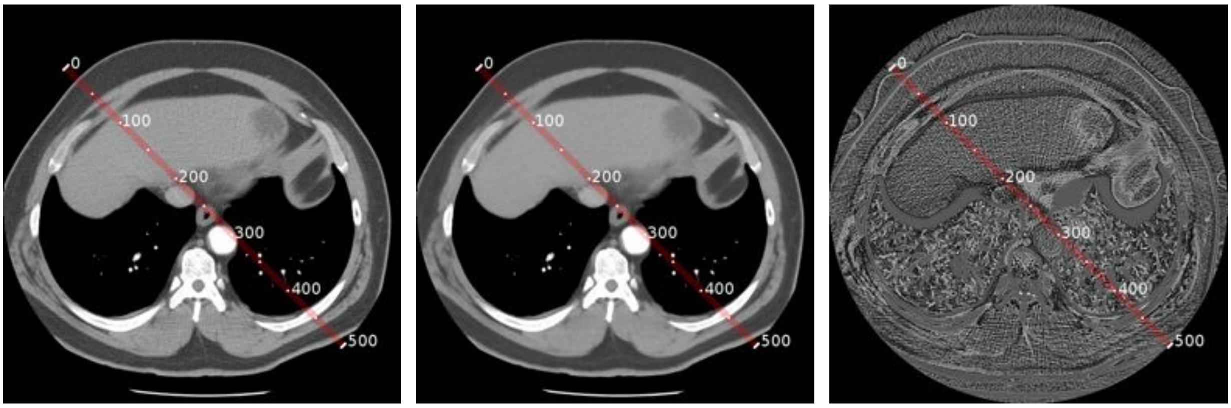


FIG. B.3: *Model of the image degradation and restoration processes.*

$$g(x,y) = h(x,y) * f(x,y) + \eta(x,y) \tag{B.1}$$

The image $f(x,y)$ is the input image before degradation by the function H together with the additive noise $\eta(x,y)$. This input image is unknown. The image $g(x,y)$ is the degraded image that is actually recorded by the imaging system, e.g. a tomographic slice such as Figure B.4(a). The image $\hat{f}(x,y)$ is an estimation of $f(x,y)$ after restoration that should be as close as possible to $f(x,y)$. Figure B.4(b) is such an estimation. Applying an intensity windowing image filter on Figure B.4(a) with a window centre 12 HU and a window width 472 HU, noise can be observed (see Appendix B.2 for details about the intensity windowing filter). Profiles corresponding to the red lines in Figure B.4 are given in Figure B.5. The corresponding profile (red curves in Figure B.5) shows high frequencies in the original data due to this noise. Therefore, a smoothing filter needs to be applied on every patient specific dataset in order to restore the data and reduce the amount of noise. An anisotropic diffusion filter has been chosen because anisotropic diffusion-based methods are iterative correction algorithms that reduce the level of noise in data but have the major advantage of preserving edges [276]. Figure B.4(b) illustrates the result of the filter on the original image. The corresponding profile (green curves in Figure B.5) shows that high frequencies have been removed from the original signal and that edges seem to have been well preserved. To further ascertain the result of the anisotropic diffusion filter, the pixelwise difference between Figures B.4(a) and B.4(b), *i.e.* the noise that has been removed from the original data, has been computed (see Figure B.4(c)). The corresponding profile (blue curves in Figure B.5) shows that i) *the noise is additive* (average ≈ 1 and standard deviation ≈ 10.2), ii) the mean of the noise signal is negligible with respect to the range of pixel intensities, this means that *the dynamic of the signal is not modified by anisotropic diffusion*, and iii) there is no overshoot in the noise signal at the spacial location of edges



(a) Original CT slice – $g(x,y)$.

(b) Restored CT slice using anisotropic diffusion – $\hat{f}(x,y)$.

(c) Noise data – $\eta(x,y)$.

FIG. B.4: *Example of anisotropic diffusion filtering on CT data (window centre = 12 HU, window width = 472 HU).*

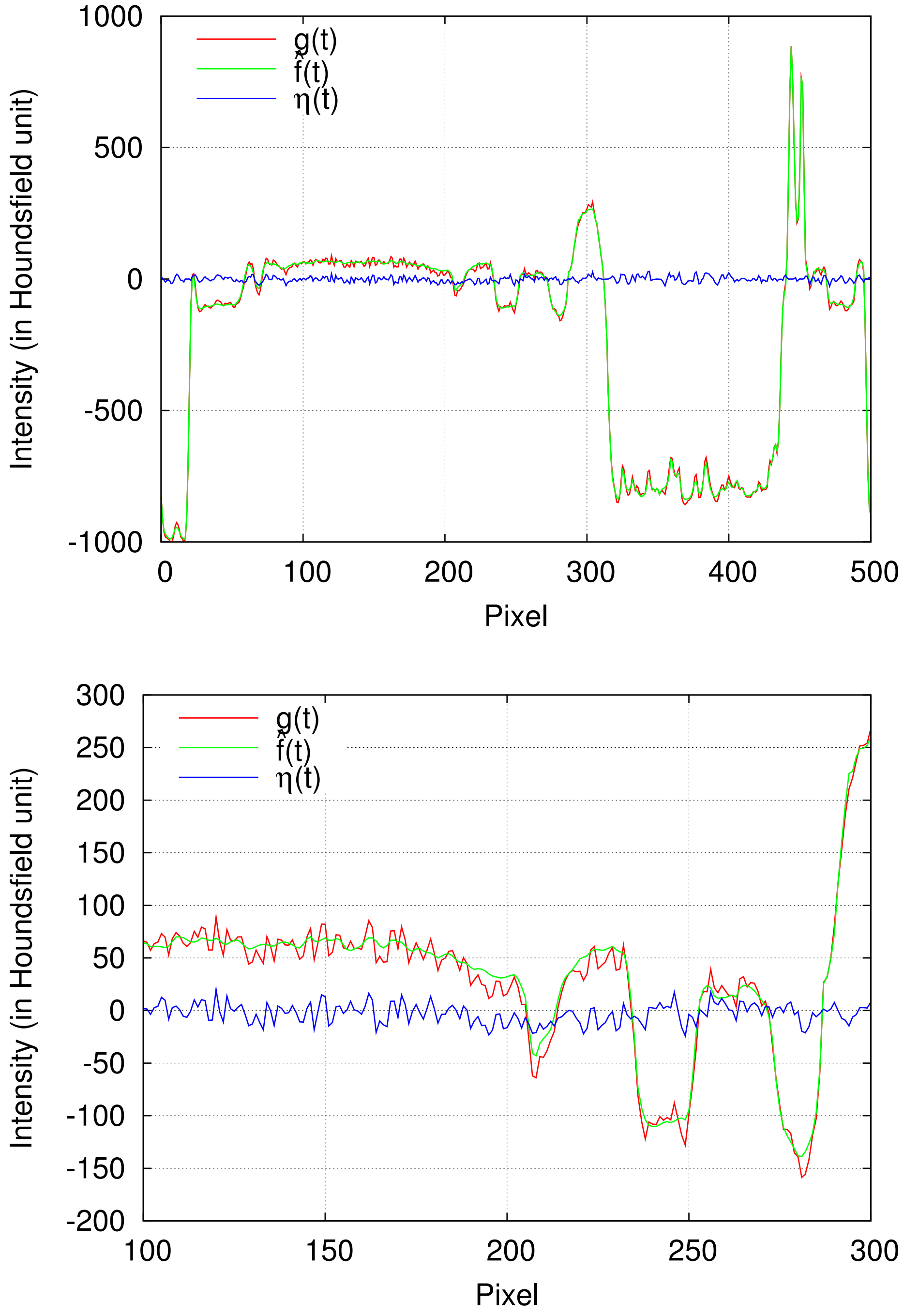


FIG. B.5: *Profiles from Figure B.4.*

in the original data, this means that *edges are well preserved*.

Normalization of the Data As stated previously, CT data is the input data for calculating i) MPR images, ii) US-like images, iii) x-ray images, iv) force feedback on the virtual image probe, and v) force feedback on the virtual needle. Images are used for guidance in real-time. Consequently, full advantage must be taken of texture hardware on modern PC graphics cards for interactive rendering of these images, and programming with framebuffer objects (FBOs), vertex and pixel shaders where appropriate. However, current hardware does not internally support textures in the CT format, *i.e.* 16 bit signed integer textures are not supported by the current generation of graphics card. If such a texture is used, the graphic hardware will convert the data to match the closest internal format that is actually supported. In this case, the developer does not have any control on such operations. To address this limitation, the data is linearly normalized between zero and one as follows:

$$Voxel_o(x,y,z) = MinValue_o + (MaxValue_o - MinValue_o) \times \frac{Voxel_i(x,y,z) - MinValue_i}{MaxValue_i - MinValue_i} \quad \text{(B.2)}$$

with $Voxel_i(x,y,z)$ the voxel of the input volume at the spatial location (x,y,z), $Voxel_o(x,y,z)$ the voxel of the output volume data at the same location, $MinValue_i$ and $MaxValue_i$ the minimum and the maximum voxel values in the input volume ($MinValue_i = -1000.0$ and $MaxValue_i = 1000.0$). $MinValue_o$ and $MaxValue_o$ the minimum and the maximum voxel values in the output volume ($MinValue_o = 0.0$ and $MaxValue_o = 1.0$). Therefore, Eq. B.2 can be shortened as follows:

$$Voxel_o(x,y,z) = \frac{Voxel_i(x,y,z) + 1000}{2000} \quad \text{(B.3)}$$

The resulting volume can then be stored in the texture memory of the graphics card using floating point format, a format that is supported by most graphics hardware. Note that, as announced at SIGGRAPH 2007, the coming version of OpenGL (Version 3.0) will add 16 and 32 bit integer internal texture formats [259].

High-pass Filtering The hybrid haptic model of needle puncture is presented in detail in Section 5.3.2. When the needle passes from one tissue type to another, a higher force can be observed. Consequently, the haptic model must take into account collisions between the tip of the needle and the surface of an organ, tissues or bones. Using volume data, it is possible to detect if the tip of the needle is located at the interface between two different surfaces using a derivative filter to compute the amplitude of the direction of the gradient at each voxel of the dataset. In digital image processing, these operations are performed as follows:

- for each pixel (i, j), the gradient is computed for two orthogonal directions. Let D_x and D_y be the gradient measurments into the x direction and the y direction respectively.
- The amplitude M of the gradient is computed as follows: $M = \sqrt{D_x^2 + D_y^2}$. To simplify computations, the amplitude is often computed as follows: $M = |D_x| + |D_y|$.
- The direction θ of the gradient is computed as follows: $\theta = \arctan(D_y/D_x)$

Figure B.6 shows how this can be modelled. The Sobel filter has proved to be efficient for such an operation [82]. Figure B.7 shows the gradient amplitude estimated using a Sobel filter on a X-ray CT slice. For this filter, Operators H_1 and H_2 are respectively:

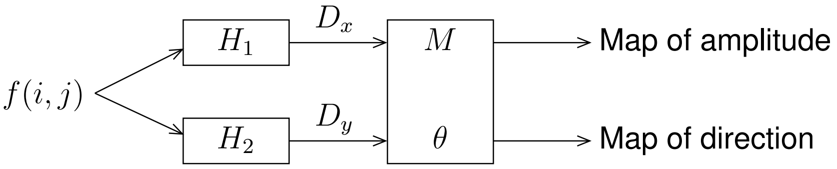


FIG. B.6: *Model of gradient estimation filter.*

$$\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

B.2 Intensity Windowing Image Filter

The intensity windowing image filter applies pixel-wise a linear transformation to the intensity values of the input image that are inside a user-defined interval. Values below this interval are changed to a constant value ($Output_{Minimum}$). Values over the interval are changed to another

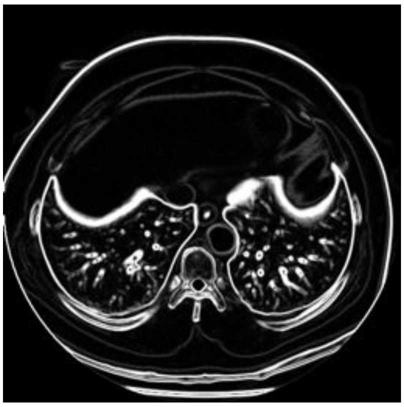


FIG. B.7: *Example of gradient estimation using a Sobel filter on a X-ray CT slice. The image corresponds to the amplitude of the gradient estimation (window centre = 2.5, window width = 5).*

constant ($Output_{Maximum}$). First of all, the user must define these constants, which then correspond to the minimum and maximum output values. Then, the window's centre and width must be specified. The linear transformation will be applied as follows:

$$Output_{Value} = Output_{Minimum} + (Output_{Maximum} - Output_{Minimum}) \times \frac{Input_{Value} - \left(Window_{Centre} - \frac{Window_{Width}}{2}\right)}{Window_{Width}} \tag{B.4}$$

where $Input_{Value}$ and $Output_{Value}$ are the input and output intensities of the current pixel respectively. $Output_{Minimum}$ and $Output_{Maximum}$ are the minimum and maximum output values respectively. $Window_{Centre}$ and $Window_{Width}$ are the window's centre and width respectively.

This operation is very common in visualization. For example, the voxels of a CT dataset stored using Hounsfield units are within the range $[-1000, 1000]$. This range is larger than what the display can output. In this case, $Output_{Minimum}$ is generally equal to zero and $Output_{Maximum}$ is generally equal to 255. By modifying the window's centre and width, the user can modify the contrast of the displayed images (see Figure B.8).

B.3 Generating Patient Specific Anatomy Models

Patient specific anatomy models are generated using VTK as follows:

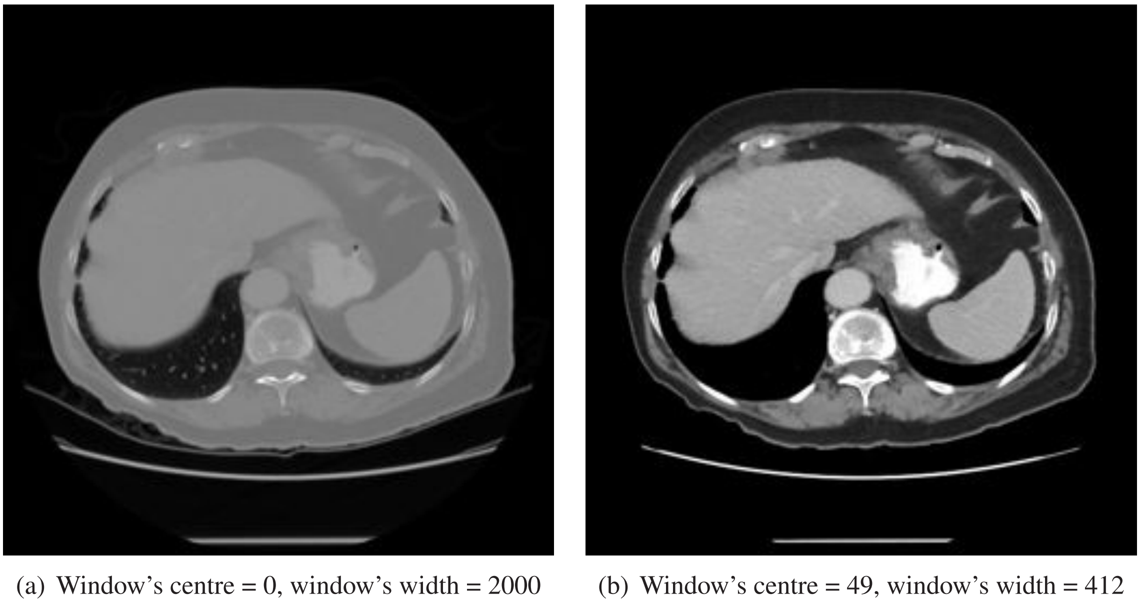

(a) Window's centre = 0, window's width = 2000 (b) Window's centre = 49, window's width = 412

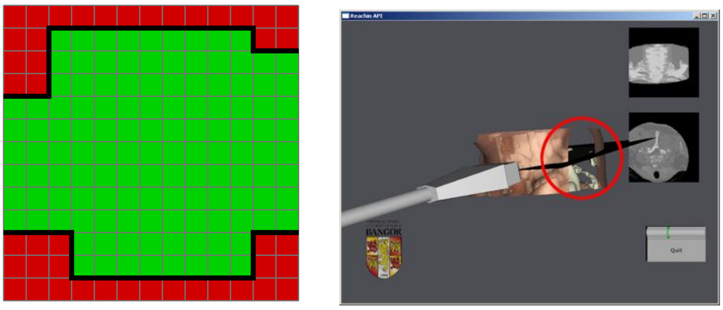
FIG. B.8: *Intensity windowing image filter applied on a CT slice.*

Read the Volume Dataset Via a graphical user interface developed using FLTK [66], the user can select the file to load. We only support 16 bit integer volumes, which is the standard format of DICOM datasets. This is also the format in which datasets are stored after the anisotropic diffusion filter has been applied.

Thresholding We assume that the user may select a volume before pixel padding removal. Therefore, threshold filters are applied once again to ensure that Hounsfield values are clamped between -1000 and 1000.

Resampling the Volume The volume data can be resampled to reduce the volume resolution so that the Marching cubes algorithm produces fewer polygons. Typically, both the width and the height of a CT slice are 512 in number of pixels, whereas the number of slices of a dataset is much lower. In our case, reducing the width and the height of slices by half and keeping the number of slices constant provides acceptable results. However, details can be lost during this processing. Therefore, the user can manually adjust the shrinking factor for each dimension separately. If the factors are equal to one, then the volume will be not resampled.

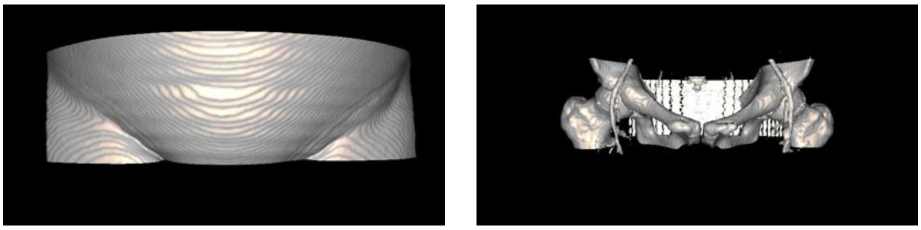
Expanding the Data The Marching cubes algorithm will generate holes if voxels corresponding to the segmented surface belong to the external boundary of the volume dataset (see Figure B.9). During clinical tests, this has proven to be a nuisance in the use of the simu-



(a) Intensity of red pixels is below iso-value, intensity of green pixels is above the iso-value, black line corresponds to the isosurface.

(b) Holes in an isosurface obtained by Marching cubes using a real X-ray CT dataset.

FIG. B.9: *Example of isosurfaces with holes that have been generated by Marching cubes.*



(a) Iso-value = -275

(b) Iso-value = 185

FIG. B.10: *Examples of isosurfaces extracted using the Marching cubes algorithm with different iso-values.*

lator. To address this, the volume dataset is simply expanded by adding, to each side of the dataset, one voxel corresponding to air.

isosurface Extraction The surface is then extracted using the Marching cubes algorithm. The user can visualize the histogram of the volume dataset and select the iso-value by manipulating using the mouse a threshold on the top of the displayed histogram. The segmented image corresponding to the iso-value will be displayed on the top the original image. Figure B.10 shows isosurfaces extracted using different iso-values.

Reducing the Number of Polygons If the number of triangles is too high, the user can decimate the polygon mesh. This is done using a multipass algorithm that deletes vertices of an existing triangle mesh depending on its local geometry and topology [229]. Vertices that pass a distance or angle criterion are removed and the holes left are filled using local triangulation. The polygon mesh in Figure B.10(a) is made of 455238 triangles. Figure B.11 shows

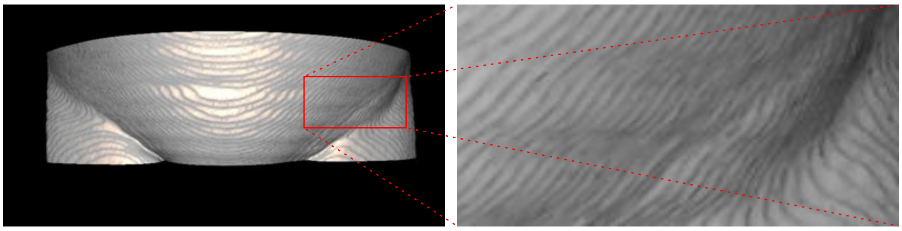


FIG. B.11: *Example of isosurface after decimation to reduce the number of triangles in the final polygon mesh.*

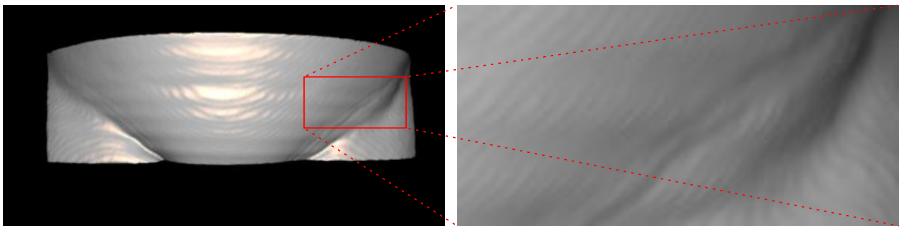


FIG. B.12: *Example of isosurface after smoothing to reduce the wood-grain effect.*

the result of the decimation filter. In this example, the number of triangles is reduced by 54%.

Smoothing the Polygon Mesh In Figures B.10(a) and B.11, artefacts can be observed in the polygon meshes. These are due to the low sampling rate of the volume dataset. This kind of artefacts is often called "wood-grain artefact". It can be attenuated by smoothing the data using a Laplacian smoothing filter that reduces high frequency information in the geometry of the triangle mesh (see Figure B.12).

Polygon reordering Finally, consistent polygon ordering is enforced and automatic determination of correct normal orientation is computed to ascertain the validity of the isosurface. This step is required to make sure that all normals are consistently oriented. A badly oriented normal might result in spurious haptic rendering.

B.4 Updating 3D Textures to track the Needle

B.4.1 Vertex program

Listing B.1 shows the GLSL vertex program that is used to set the texture coordinates of the current vertex:

LISTING B.1: *Vertex program used to set the texture coordinates of the current vertex.*

```
void main()
{
  /* Setup the texture coordinates of the current
   * vertex, i.e. its 3D location with respect
   * to the volume dataset (3D texture) coordinate
   * space:
   */

  // NOTE: we use untransformed position as
  // texture coordinates:
  gl_TexCoord[0] = gl_MultiTexCoord0;
  gl_TexCoord[1] = gl_MultiTexCoord1;

  // Setup the position of the current vertex,
  // i.e. its 2D location in screen:
  gl_Position = ftransform();
}
```

B.4.2 Fragment program

The boolean function **IsPositionIntoNeedle(const vec3 aPosition)** has been implemented in a GLSL fragment program to detect whether a voxel at a given texture coordinates has been penetrated by the needle (see Listing B.2).

LISTING B.2: *Fragment program used to detect whether a voxel at a given texture coordinates has been penetrated by the needle.*

```
uniform sampler3D CtTexture;     // The CT dataset
uniform vec3 VolumeSize;         // The CT datasset size [in mm]

uniform vec2 NeedleProperties;   // NeedleProperties[0] = Radius
                                 // NeedleProperties[1] = Length
```

```

uniform vec3 NeedlePosition;   // Needle position
uniform vec3 NeedleDirection;  // Needle direction

// Check if the current pixel, corresponding to ‘‘aPosition’’ [in mm],
// is into the needle.
bool IsPositionIntoNeedle(const vec3 aPosition);


void main()
{
  float intensity = 0.0;

  // The texel belongs to the needle
  if (IsPositionIntoNeedle(vec3(gl_TexCoord[0].s * VolumeSize.x,
    gl_TexCoord[0].t * VolumeSize.y,
    gl_TexCoord[0].p * VolumeSize.z)))
  {
    intensity = 1.0;
  }
  // The texel is not into the needle
  // Test if the texture coordinates are valid
  else if (0.0 <= gl_TexCoord[0].s && gl_TexCoord[0].s <= 1.0 &&
      0.0 <= gl_TexCoord[0].t && gl_TexCoord[0].t <= 1.0 &&
      0.0 <= gl_TexCoord[0].p && gl_TexCoord[0].p <= 1.0)
  {
    intensity = texture3D(CtTexture, gl_TexCoord[0].stp).r;
  }

  gl_FragColor = vec4(intensity, intensity, intensity, 1.0);
}


bool IsPositionIntoNeedle(const vec3 aPosition)
{
  vec3 NV = aPosition - NeedlePosition;
  vec3 UV = aPosition - (NeedlePosition -
    NeedleDirection * NeedleProperties[1]);

  // Outsite of the needle (front of the needle)
  float dot_product = dot(NeedleDirection, NV);
  if (dot_product > 0.0) return false;

  // Outsite of the needle (behind the needle)
  if (dot(NeedleDirection, UV) < 0.0) return false;

  // Inside needle
  float length_NV = length(NV);
  float square_length_VW = length_NV * length_NV -
    dot_product * dot_product;
  if (square_length_VW > pow(NeedleProperties[0], 2.0))
    return false;
  else
    return true;
}
```

B.5 Updating 3D Textures to add a Lesion into the Patient

The boolean function **IsPositionIntoLesion(const vec3 aPosition)** has been implemented in a GLSL fragment program to detect whether a voxel at a given texture coordinate is located within the lesion (see Listing B.3).

LISTING B.3: *Fragment program used to detect whether a voxel at a given texture coordinates is located into the lesion.*

```
uniform sampler3D CtTexture;    // The CT dataset
uniform vec3 VolumeSize;        // The CT datasset size [in mm]

uniform vec3  TargetPosition;    // Position of the centre of
                                 // the target [in mm]
uniform vec2  TargetProperties; // TargetProperties[0] = Radius
                                 // TargetProperties[1] = Voxel value

// Check if the current pixel, corresponding to "aPosition" [in mm],
// is into the target.
bool IsPositionIntoTarget(const vec3 aPosition);

void main()
{
  float intensity = 0.0;

  // Check if the texel is into the target
  if (IsPositionIntoTarget(pixel_position))
  {
    intensity = TargetProperties[1];
    is_into_needle = 0.9;
  }

  // The texel is not into the target
  // Test if the texture coordinates are valid
  else if (0.0 <= gl_TexCoord[0].s && gl_TexCoord[0].s <= 1.0 &&
      0.0 <= gl_TexCoord[0].t && gl_TexCoord[0].t <= 1.0 &&
      0.0 <= gl_TexCoord[0].p && gl_TexCoord[0].p <= 1.0)
  {
    intensity = texture3D(CtTexture, gl_TexCoord[0].stp).r;
  }

  gl_FragColor = vec4(intensity, intensity, intensity, 1.0);
}

bool IsPositionIntoTarget(const vec3 aPosition)
{
  // Inside of the target
  if (distance(aPosition, TargetPosition) < TargetProperties[0])
  {
    return true;
  }
  // Outside of the target
  else
  {
    return false;
  }
}
```

B.6 Needle Tracking in Ultrasound Images

Listing B.4 is an excerpt of the fragment program that is used to compute the 2D MPR image, to track the needle, and to eventually add a lesion. The source code of the boolean function **IsPositionIntoNeedle(const vec3 aPosition)** and **IsPositionIntoTarget(const vec3 aPosition)** are available in Listings B.2 and B.3 respectively.

LISTING B.4: *Fragment program used to detect whether a voxel at a given texture coordinates has been penetrated by the needle.*

```
uniform sampler3D CtTexture;      // The CT dataset
uniform sampler2D USMaskTexture; // The mask for US image

uniform int IsLinearArray;       // 1 if linear array
```

```
                                  // 0 if curvilinear array

uniform vec3 VolumeSize;          // The CT dataset size in mm

uniform int Puncturing;   // 0 if the needle is not into the body
                          // 1 if the needle is not into the body

uniform vec2 NeedleProperties;    // NeedleProperties[0] = Diameter;
                                  // NeedleProperties[1] = Length

uniform vec3 NeedlePosition;      // Needle position
uniform vec3 NeedleDirection;     // Needle direction

uniform int   IsTargetEnable;     // 0 if the target is not enable
                                  // 1 if the target is enable
uniform vec3  TargetPosition;
uniform vec2  TargetProperties;   // TargetProperties[0] = Radius
                                  // TargetProperties[1] = Voxel value


//  Check if the current pixel, corresponding to "aPosition",
//  is into the needle.
bool IsPositionIntoNeedle(const vec3 aPosition);



//  Check if the current pixel, corresponding to "aPosition",
//  is into the target.
bool IsPositionIntoTarget(const vec3 aPosition);


void main()
{
    float intensity = 0.0;
    float is_into_needle = 0.0;


  // Test if linear array, if not test the mask
  if (IsLinearArray == 1 ||
    (IsLinearArray == 0 &&
      texture2D(USMaskTexture, gl_TexCoord[1].st).r >= 0.5))
    // Test the texture coordinates
    if (0.0 <= gl_TexCoord[0].s && gl_TexCoord[0].s <= 1.0 &&
        0.0 <= gl_TexCoord[0].t && gl_TexCoord[0].t <= 1.0 &&
        0.0 <= gl_TexCoord[0].p && gl_TexCoord[0].p <= 1.0)
  {
    vec3 pixel_position = vec3(gl_TexCoord[0].s * VolumeSize.x,
      gl_TexCoord[0].t * VolumeSize.y,
      gl_TexCoord[0].p * VolumeSize.z);
    intensity = texture3D(CtTexture, gl_TexCoord[0].stp).r;

    // A virtual lesion has been added
    if (IsTargetEnable != 0)
    {
      // Check if the texel is into that lesion
      if (IsPositionIntoTarget(pixel_position))
      {
        intensity = TargetProperties[1];
        is_into_needle = 0.9;
      }
    }

    // The needle is inserted into the body
    if (Puncturing == 1)
    {
      // Check if the texel is into the needle
      if (IsPositionIntoNeedle(pixel_position))
      {
        //intensity = 1.0;
        is_into_needle = 1.0;
      }
    }
  }

  gl_FragColor = vec4(intensity, is_into_needle, intensity, 1.0);
}
```

B.7 Interactive Determinist X-Ray Simulation on the GPU

B.7.1 Ray definition

A ray can be represented as a straight line l passing through two points, as P_1 and P_2 in Figure B.13, with $\mathbf{OP_1}$ and $\mathbf{OP_2}$, which can also be written $\mathbf{p_1}$ and $\mathbf{p_2}$, the position vectors of P_1 and P_2 respectively. This line, as any other line, can be infinitely long. However, the ray may be used as the line segment P_1P_2. Let P_3 be a point located on the straight line passing through points P_1

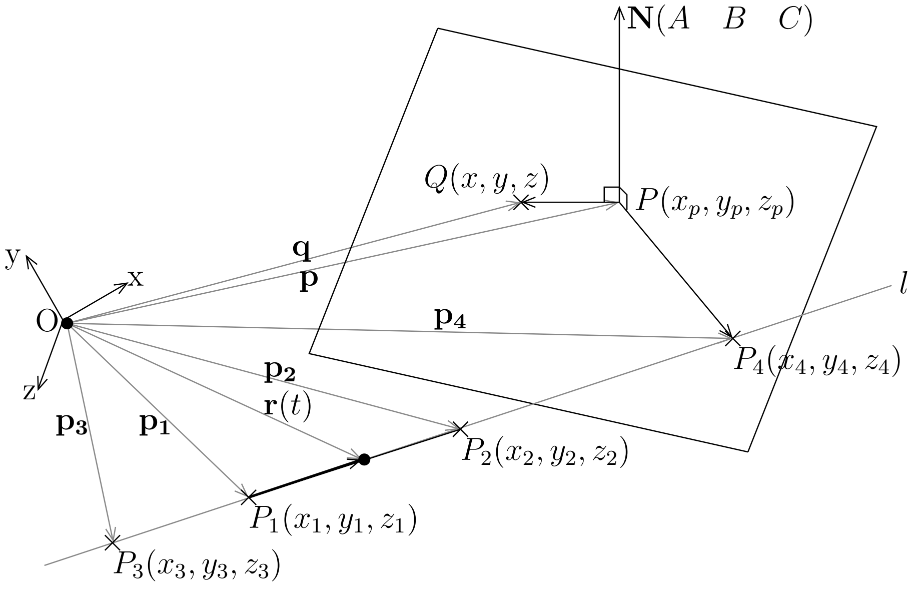


FIG. B.13: *Intersection ray-plane.*

and P_2. The position vector of point P_3 is $\mathbf{p_3}$. Thus we have:

$$\mathbf{p_3} = \mathbf{p_1} + t \times (\mathbf{p_2} - \mathbf{p_1}) \tag{B.5}$$

t varies to give all position vectors of P_3 at any location on the line l. If t is between 0 and 1, then P_3 will be located on the line segment between P_1 and P_2; else if t is negative, then P_3 will be at the front of P_1 as it is the case in Figure B.13; else P_3 will be after P_2. In Eq. B.5, $\mathbf{p_3}$ can be replaced by '$\mathbf{r}(t)$', a variable vector depending on t. It gives Eq. B.6, the vector equation of the straight line passing through points P_1 and P_2, *i.e.* the equation of a ray:

$$\mathbf{r}(t) = \mathbf{p_1} + t \times (\mathbf{p_2} - \mathbf{p_1}) \tag{B.6}$$

B.7.2 Box definition

"Generalized boxes" are important in computer graphics because they can be used as efficient bounding volumes [126, 273]. Indeed, hierarchical representation of bounding volumes are often used in computer graphics to avoid unnecessary computations, particularly in ray tracing of complex scenes. A generalized box is formed from pairs of parallel planes, the area between a pair of planes is called a slab [126]. Although pairs of planes can be at any angle with respect to each other, we only consider the special case of rectangular parallelepipeds made of three pairs of planes as described by Rubin and Whitted [214] and oriented as in Figure B.14.

A plane can be defined by a point located onto it, and a vector normal to that plane. Let $P(x_p, y_p, z_p)$ be a point located onto the plane whose normal is $\mathbf{N}(A \quad B \quad C)$. As illustrated in Figure B.13, any point $Q(x,y,z)$ is also located onto the plane if the vector $\mathbf{PQ}((x-x_p) \quad (y-y_p) \quad (z-z_p))$ is perpendicular to the normal **N**, *i.e.* the dot product between **PQ** and **N** is null:

$$\begin{aligned}
\mathbf{N}.\mathbf{PQ} &= 0 && \text{(B.7)}\\
A\times(x-x_p)+B\times(y-y_p)+C(z-z_p) &= 0 && \text{(B.8)}\\
A\times x+B\times y+C\times z-(A\times x_p+B\times y_p+C\times z_p) &= 0 && \text{(B.9)}\\
A\times x+B\times y+C\times z+D &= 0 && \text{(B.10)}
\end{aligned}$$

Eq. B.10 gives the equation of the plane with $D = -(A\times x_p + B\times y_p + C\times z_p)$.

For efficiency reasons, the parallelepiped boxes will never rotate. Indeed, it will be computationally intensive to rotate each voxel of a high resolution CT data set. Then, as illustrated by Figure B.14, a box can be defined by its lower corner and its upper corner, the points $P_{top}(x_{top}, y_{top}, z_{top})$ and $P_{bottom}(x_{bottom}, y_{bottom}, z_{bottom})$. For each pair of parallel plane, the normal $\mathbf{N}(A \quad B \quad C)$ is given by:

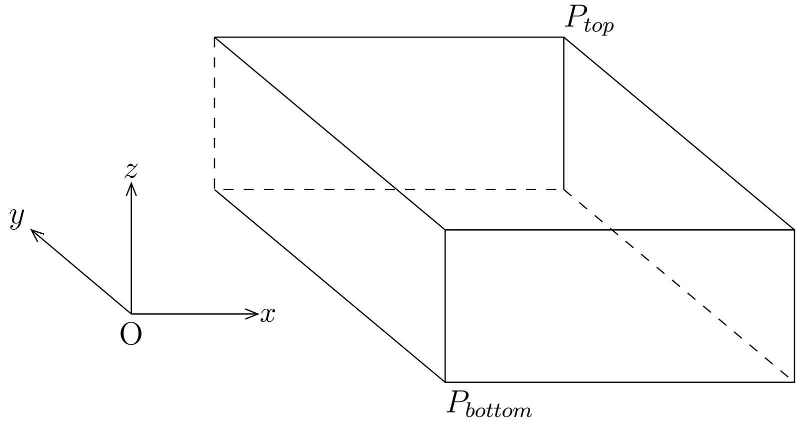


FIG. B.14: *Cube orientation.*

- A = 1, B = 0, C = 0 for the two planes parallel to the *y*-*z* plane;
- A = 0, B = 1, C = 0 for the two planes parallel to the *x*-*z* plane;
- A = 0, B = 0, C = 1 for the two planes parallel to the *x*-*y* plane.

B.7.3 Intersection computations

Let P_4 be a point located onto the plane. The ray is intersecting the plane in P_4 if the following equation system is fulfilled:

$$\begin{cases} A \times x_4 + B \times y_4 + C \times z_4 + D = 0 \\ \mathbf{p_4} = \mathbf{p_1} + t \times (\mathbf{p_2} - \mathbf{p_1}) \end{cases} \tag{B.11}$$

We can deduce t from Eq. B.11, it gives:

$$t = -\frac{A \times x_1 + B \times y_1 + C \times z_1 + D}{A \times (x_2 - x_1) + B \times (y_2 - y_1) + C \times (z_2 - z_1)} \tag{B.12}$$

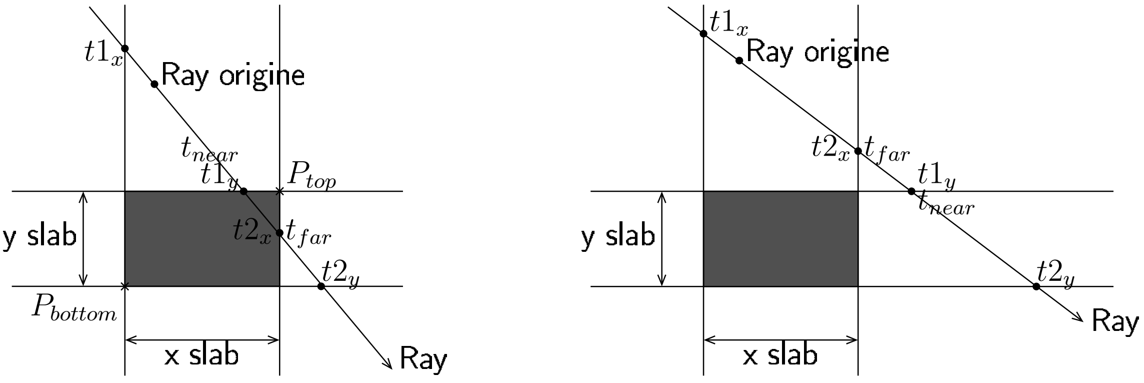
Note that if t is between 0 and 1, then the intersection of the ray with plane will be between P_1 and P_2; else if t is negative, then the intersection will be at the front of P_1; else the intersection will be at the rear of P_2 as it is shown in Figure B.13.

As stated previously, the vector normal to the planes parallel to the y-z plane is $\mathbf{N}(1 \quad 0 \quad 0)$. The intersections of the ray with this pair of planes passing through P_{top} and P_{bottom} are given respectively by:

$$t_{top\_x} = \frac{(x_{top} - x_1)}{(x_2 - x_1)}$$

$$t_{bottom\_x} = \frac{(x_{bottom} - x_1)}{(x_2 - x_1)}$$

As shown by Figure B.15, $t1_x$ and $t2_x$ are the intersections with the closest and the furthest planes respectively. The computations for t_{top_y}, t_{bottom_y}, $t1_y$, $t2_y$, t_{top_z}, t_{bottom_z}, $t1_z$ and $t2_z$ are similar. As shown in 2D by Figure B.15(a), the largest value of $t1$ gives t_{near}, the entry point of the ray into the box, and the smallest value of $t2$ gives t_{far}, the exit point of the ray into the box. As shown in Figure B.15(b), if t_{far} is smaller than t_{near}, then the ray misses the box. The algorithm implemented to compute intersections between the ray and a parallelepiped box is available in Figure B.16 in Appendix B.7.4. The following section describes how the algorithm has been implemented.



(a) (b)

FIG. B.15: *Intersection of a ray with a box.*

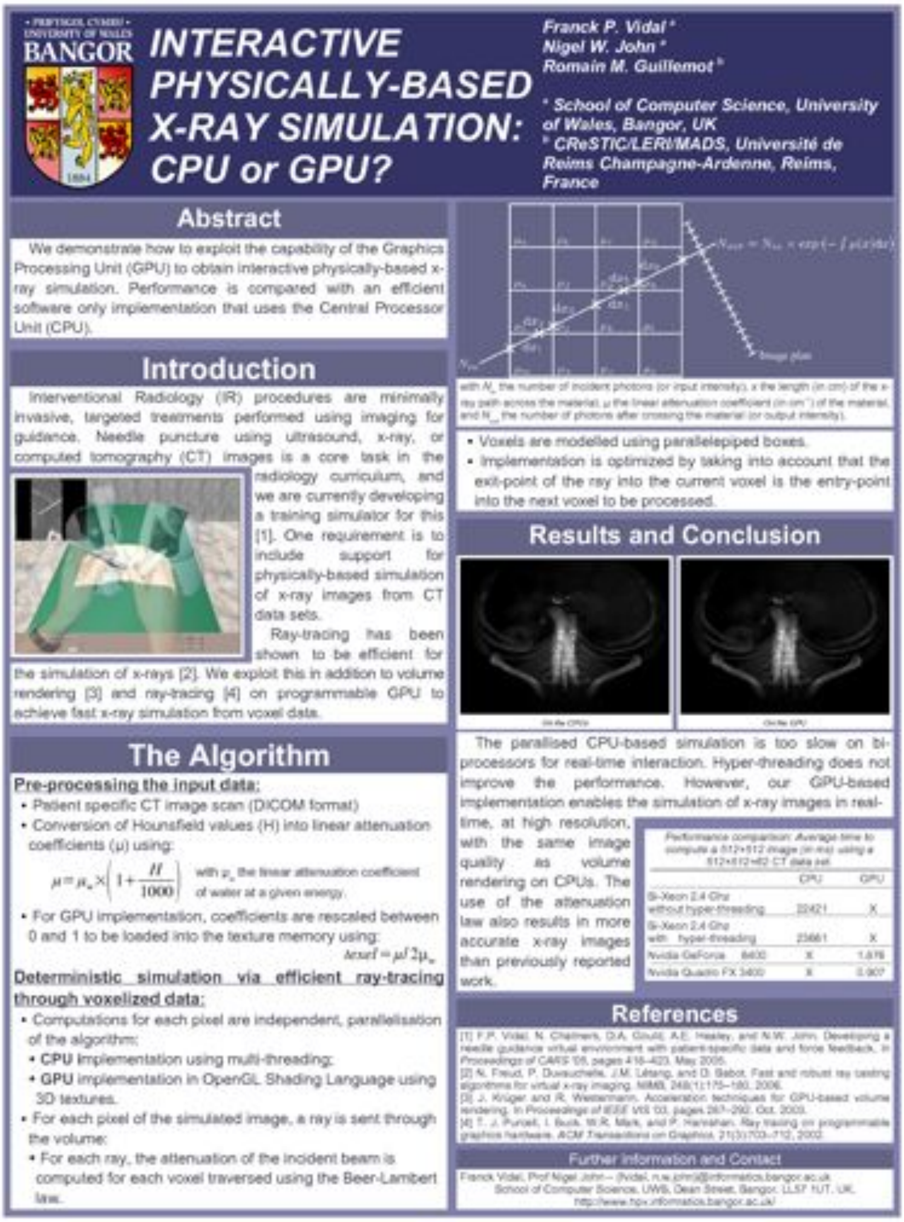
B.7.4 Intersection Ray-Box

The algorithm presented in Figure B.16 is used to compute intersections between a ray and a parallelepiped box.

```
input  : $P_1$ the start point of the ray, $P_2$ the end point of the ray, $P_{bottom}$, $P_{top}$ the lower and upper
         corners of the box respectively.
output : If the ray intersects the box then $t_{near}$ and $t_{far}$ are the entry and the exit points of the ray
         into the box respectively.
return : “true” if the ray intersects the box, “false” otherwise.
1  begin
       /* Initialisation                                                        */
2      $t_{near} \longleftarrow -\infty$
3      $t_{far} \longleftarrow +\infty$
       /* Intersections with $y$-$z$ plane                                      */
4      if $(x_2 - x_1)$ is null then                /* Ray parallel to the $y$-$z$ plane */
5          if $x_1 < x_{bottom}$ and $x_1 > x_{top}$ then return false;        /* No intersection */
6      else
7          $t_1 \longleftarrow (x_{bottom} - x_1)/(x_2 - x_1)$          /* intersection with near plane */
           $t_2 \longleftarrow (x_{top} - x_1)/(x_2 - x_1)$          /* $t_2$ intersection with far plane */
8          if $(t_1 > t_2)$ then SWAP($t_1, t_2$); /* Since $t_1$ intersection with near plane and $t_2$
           intersection with far plane */
9          if $t_1 > t_{near}$ then $t_{near} \longleftarrow t_1$;                      /* Get largest $t_{near}$ */
10         if $t_2 < t_{far}$ then $t_{far} \longleftarrow t_2$;                      /* Get smallest $t_{far}$ */
11         if $t_{near} > t_{far}$ then return false;                          /* No intersection */
12         if $t_{far} < 0$ then return false;                  /* Intersection behind observer */
13     endif
14     The computations for the $x$-$z$ plane and the $x$-$y$ plane are similar. Just repeat the
       instructions twice from line 4 to line 13; replacing “X” by “Y” and “Z” respectively.
15     return true
16 end
```

FIG. B.16: *Intersection Ray-Box*

B.7.5 Poster at MMVR



• PRIFYSGOL CYMRU • UNIVERSITY OF WALES BANGOR

INTERACTIVE PHYSICALLY-BASED X-RAY SIMULATION: CPU or GPU?

Franck P. Vidal [a]
Nigel W. John [a]
Romain M. Guillemot [b]

[a] School of Computer Science, University of Wales, Bangor, UK
[b] CReSTIC/LERI/MADS, Université de Reims Champagne-Ardenne, Reims, France

Abstract

We demonstrate how to exploit the capability of the Graphics Processing Unit (GPU) to obtain interactive physically-based x-ray simulation. Performance is compared with an efficient software only implementation that uses the Central Processor Unit (CPU).

Introduction

Interventional Radiology (IR) procedures are minimally invasive, targeted treatments performed using imaging for guidance. Needle puncture using ultrasound, x-ray, or computed tomography (CT) images is a core task in the radiology curriculum, and we are currently developing a training simulator for this [1]. One requirement is to include support for physically-based simulation of x-ray images from CT data sets.

Ray-tracing has been shown to be efficient for the simulation of x-rays [2]. We exploit this in addition to volume rendering [3] and ray-tracing [4] on programmable GPU to achieve fast x-ray simulation from voxel data.

The Algorithm

Pre-processing the input data:

- Patient specific CT image scan (DICOM format)
- Conversion of Hounsfield values (H) into linear attenuation coefficients (μ) using:

$$\mu = \mu_w \times \left(1 + \frac{H}{1000}\right)$$ with μ_w the linear attenuation coefficient of water at a given energy.

- For GPU implementation, coefficients are rescaled between 0 and 1 to be loaded into the texture memory using:

$$texel = \mu / 2\mu_w$$

Deterministic simulation via efficient ray-tracing through voxelized data:

- Computations for each pixel are independent, parallelisation of the algorithm:
  - **CPU** implementation using multi-threading;
  - **GPU** implementation in OpenGL Shading Language using 3D textures.
- For each pixel of the simulated image, a ray is sent through the volume:
  - For each ray, the attenuation of the incident beam is computed for each voxel traversed using the Beer-Lambert law.

with N_{in} the number of incident photons (or input intensity), x the length (in cm) of the x-ray path across the material, μ the linear attenuation coefficient (in cm^{-1}) of the material, and N_{out} the number of photons after crossing the material (or output intensity).

- Voxels are modelled using parallelepiped boxes.
- Implementation is optimized by taking into account that the exit-point of the ray into the current voxel is the entry-point into the next voxel to be processed.

Results and Conclusion

On the CPUs

On the GPU

The parallised CPU-based simulation is too slow on bi-processors for real-time interaction. Hyper-threading does not improve the performance. However, our GPU-based implementation enables the simulation of x-ray images in real-time, at high resolution, with the same image quality as volume rendering on CPUs. The use of the attenuation law also results in more accurate x-ray images than previously reported work.

Performance comparison: Average time to compute a 512×512 image (in ms) using a 512×512×82 CT data set.

	CPU	GPU
Bi-Xeon 2.4 Ghz without hyper-threading	22421	X
Bi-Xeon 2.4 Ghz with hyper-threading	23661	X
Nvidia GeForce 6600	X	1.876
Nvidia Quadro FX 3400	X	0.907

References

[1] F.P. Vidal, N. Chalmers, D.A. Gould, A.E. Healey, and N.W. John. Developing a needle guidance virtual environment with patient-specific data and force feedback. In *Proceedings of CARS '05*, pages 418–423, May 2005.
[2] N. Freud, P. Duvauchelle, J.M. Létang, and D. Babot. Fast and robust ray casting algorithms for virtual x-ray imaging. NIMB, 248(1):175–180, 2006.
[3] J. Krüger and R. Westermann. Acceleration techniques for GPU-based volume rendering. In *Proceedings of IEEE VIS '03*, pages 287–292, Oct. 2003.
[4] T. J. Purcell, I. Buck, W.R. Mark, and P. Hanrahan. Ray tracing on programmable graphics hardware. *ACM Transactions on Graphics*, 21(3):703–712, 2002.

Further information and Contact

Franck Vidal, Prof Nigel John -- {fvidal, n.w.john}@informatics.bangor.ac.uk
School of Computer Science, UWB, Dean Street, Bangor, LL57 1UT, UK.
http://www.hpv.informatics.bangor.ac.uk/

FIG. B.17: *Poster presented at MMVR 15, 6-9 February 2007, Long Beach, California, USA.*

Appendix C

EVALUATION PROCESS

C.1 Questionnaire 1

Thank you for taking time to fill out this questionnaire. Please answer the following items by circling the alternative that you agree most.

1)	The visual rendering is good enough for the training.						
	Strongly disagree	Moderately disagree	Slightly disagree	Don't know	Slightly agree	Moderately agree	Strongly agree
2)	The visual rendering is good enough for the training.						
	Strongly disagree	Moderately disagree	Slightly disagree	Don't know	Slightly agree	Moderately agree	Strongly agree
3)	The visual rendering is good enough for the training.						
	Strongly disagree	Moderately disagree	Slightly disagree	Don't know	Slightly agree	Moderately agree	Strongly agree
4)	The visual rendering is good enough for the training.						
	Strongly disagree	Moderately disagree	Slightly disagree	Don't know	Slightly agree	Moderately agree	Strongly agree
5)	The visual rendering is good enough for the training.						
	Strongly disagree	Moderately disagree	Slightly disagree	Don't know	Slightly agree	Moderately agree	Strongly agree

Further comments?

TAB. C.1: *Evaluation questionnaire used at the Interventional Radiology Study Day 2005, Preston Hospital, July 19th 2005.*

C.2 Questionnaire 2

Questionnaire: completed at each attendance: circle as appropriate.

1. SpR / fellow / consultant
2. Male / female.
3. Age: ____yrs
4. Number of years in radiology ________(state number)
5. How confident do you feel when performing imaging guided needle puncture procedures?

no confidence: I can't do these procedures			some confidence: I can perform these under supervisor guidance			full confidence: I can perform unsupervised in all cases			
1	2	3	4	5	6	7	8	9	10

(circle level of confidence).

6. If an SpR / fellow: how many logbook interventional procedures have you performed during the past 3 months? _________ (state number).
   How many of these were performed independently?_________ (state number).
7. How similar did you find the procedure on the simulator model to the same kind of procedure on a patient?

very different	fairly different	neutral	fairly similar	very similar
1	2	3	4	5

8. **How much do you agree or disagree with the following statements:**
   (1=strongly disagree, 2=somewhat disagree, 3=acceptable, 4=somewhat agree, 5=strongly agree)

   - **Simulation of the patient environment:**
     (a) CT visual elements were of high quality 1 – 2 – 3 – 4 – 5
     (b) Tactile (touch) elements were accurate 1 – 2 – 3 – 4 – 5
   - **Simulation of the needle insertion procedure:**
     (c) Needle insertion through skin is realistic 1 – 2 – 3 – 4 – 5
     (d) Needle insertion through tissues 1 – 2 – 3 – 4 – 5
   - **General:**
     (e) The level of difficulty of the cases was appropriate 1 – 2 – 3 – 4 – 5
     (f) The cases covered situations encountered in my practice 1 – 2 – 3 – 4 – 5
     (g) The cases were a suitable challenge for my skills 1 – 2 – 3 – 4 – 5
     (h) The simulator reflected a high level of realism 1 – 2 – 3 – 4 – 5

9. Further comments
   (on next page)

TAB. C.2: *Evaluation questionnaire used at the Interventional Radiology Study Day 2006, Preston Hospital, July 31st 2007*

C.3 Questionnaire 3

Questionnaire: completed at each attendance: circle as appropriate.

1. SpR / fellow / consultant
2. Male / female.
3. Age: ____yrs
4. Number of years in radiology ________(state number)
5. How confident do you feel when performing imaging guided needle puncture procedures?

   no confidence: I can't do these procedures — some confidence: I can perform these under supervisor guidance — full confidence: I can perform unsupervised in all cases

   1 2 3 4 5 6 7 8 9 10

   (circle level of confidence).
6. If an SpR / fellow: how many logbook interventional procedures have you performed during the past 3 months? ________ (state number).
   How many of these were performed independently?________ (state number).
7. How similar did you find the procedure on the simulator model to the same kind of procedure on a patient?

very different	fairly different	neutral	fairly similar	very similar
1	2	3	4	5

8. **How much do you agree or disagree with the following statements:**
   (1=strongly disagree, 2=somewhat disagree, 3=acceptable, 4=somewhat agree, 5=strongly agree)

   - **Simulation of the patient environment:**
     (a) US visual elements were of high quality 1 – 2 – 3 – 4 – 5
     (b) Tactile (touch) elements were accurate 1 – 2 – 3 – 4 – 5
   - **Simulation of the needle insertion procedure:**
     (c) Needle insertion through skin is realistic 1 – 2 – 3 – 4 – 5
     (d) Needle insertion through tissues 1 – 2 – 3 – 4 – 5
   - **General:**
     (e) The level of difficulty of the cases was appropriate 1 – 2 – 3 – 4 – 5
     (f) The cases covered situations encountered in my practice 1 – 2 – 3 – 4 – 5
     (g) The cases were a suitable challenge for my skills 1 – 2 – 3 – 4 – 5
     (h) The simulator reflected a high level of realism 1 – 2 – 3 – 4 – 5
9. Further comments
   (on next page)

TAB. C.3: *Evaluation questionnaire used at the Interventional Radiology Study Day 2007, Preston Hospital, July 16th 2007*

Glossary

A

acoustic impedance Property of medium causing resistance to the propagation of ultrasound. It is defined as $Z = r \times c$, with r the tissue density, and c the velocity of the ultrasound wave in the medium. It is measured in Rayls.

algebraic reconstruction technique (ART) Discrete tomographic reconstruction methods based on iterative correction algorithms.

alternating current (AC) Type of electric current.

aneurysm Blood-filled dilatation of a blood vessel caused by a disease or weakening of the vessel's wall.

angiography Examination of the blood vessels using x-rays following the injection of a contrast agent.

angioplasty Procedure consisting of widening a narrow or blocked blood vessel.

apodization Method for reducing side lobes. In practice, it gradually decreases the vibration of the transducer surface with distance from its center.

application programming interface (API) Provide an interface between a programming language and lower level utilities.

artefact Streaks, spots, blurring, *etc.* of a picture obscuring details and/or showing things that "should not really be there".

arteriogram Radiogram produced by arteriography.

arteriography Examination of an artery using x-rays following the injection of a contrast agent.

B

balloon Bag that is inserted into a body cavity or structure and inflated with air or gas.

Bangor Image Guided NEedle Puncture SImulator (BIGN*e*PS*i*) Virtual environment for training ultrasound guided image puncture (developed by Franck P. Vidal at the University of Wales, Bangor).

British Society of Interventional Radiologists (BSIR) See `<http://www.bsir.org/>` (accessed 11th October 2006).

C

C for Graphics (Cg) Is a proprietary high level shading language developed by Nvidia to provide more direct control of the graphics pipeline. This language is based on the C programming language.

calculus (*plural* calculi) (also called a stone) a solid mass formed in the gallbladder, kidneys, or urinary bladder.

Calman system Reforms proposed by K. Calman to restructure the training of middle grade medical staff to produce a shorter, more structured and organised training than in the past in many disciplines.

Cardiovascular and Interventional Radiological Society of Europe (CIRSE) See `<http://www.cirse.org/>` (accessed 11th October 2006).

catheter Plastic tube.

cathode-ray tube (CRT) vacuum glass envelope containing a cathode that emits electrons. Electrons are accelerated and deflected to hit a fluorescent screen and emit light.

Center for the Integration of Medicine and Innovative Technology (CIMIT) See `<http://www.cimit.org/>` (accessed 11th October 2006).

central processing unit (CPU) Often simply called processor. It is the component in a digital computer that interprets instructions and processes data.

central venous catheter (CVC) Medical procedure in which a catheter is placed into a large vein in the neck ().

charge-coupled device (CCD) Image sensor consisting of an integrated circuit containing an array of linked, or coupled, capacitors sensitive to light.

Chiba needle Hollow needle used in interventional radiology, named after Chiba University in Japan, where it was invented.

cognitive task analysis (CTA) Study which aims to identify, describe and detail the whole process of a procedure, from the cues perceived, to the cognitive thought processes of decision-making, to the ensuing physical actions.

COLLAborative Design Activity (COLLADA) Open file format designed to describe 3D data. It aims is to ease the exchange between multiple software packages. COLLADA uses using an XML syntax.

Collaborators in Radiological Interventional Virtual Environments (CR*a*IVE) See <http://www.craive.org.uk/> (accessed 11$^{\text{th}}$ October 2006).

Compute Unified Device Architecture (CUDA) Is a GPGPU technology developed by Nvidia to use the C programming language to code algorithms for execution on the GPU.

computer-aided design (CAD) Computer-based tools to assist engineers, architects, industrial designers, *etc.* in their design activities.

Computer Haptics & Active Interfaces (CHAI 3D) Is an open source toolkit written in C++ for computer haptics, visualization and interactive real-time simulation.

computerized axial tomography (CAT) Generally reduced as computerized tomography. See Computerized Tomography.

computerized tomography (CT) Also called computed tomography, originally known as Computerized Axial Tomography. X-ray CT is the most commonly used imaging modality in hospital to routinely acquire 3D data sets from patients.

© copyright.

curved planar reconstruction (CPR) Volume rendering technique based on a “curved” slice following the trajectory of a blood vessel.

D

degrees of freedom (DOF) Number of dimensions in which a haptic device can either be tracked, or provide force feedback.

Digital Imaging and COmmunications in Medicine (DICOM) Set of standards for handling, storing, printing, and transmitting information in medical imaging.

direct current (DC) Type of electric current.

E

echo-planar imaging (EPI) One of the magnetic resonance imaging techniques.

electrocardiogram (ECG) Graphical recording of the cardiac cycle produced by an electrocardiograph, a device that measures electrical potentials on the body surface.

embolisation Occlusion of a blood vessel using a coil.

endograft Graft introduced using endoscopy.

Engineering and Physical Sciences Research Council (EPSRC) UK Government's leading funding agency for research and training in engineering and the physical sciences.

extensible markup language (XML) A W3C-recommended general-purpose markup language that supports a wide variety of applications. XML is a simplified subset of Standard Generalized Markup Language (SGML).

F

Fast Light Toolkit (FLTK) Is pronounced "fulltick". It is a cross-platform C++ GUI toolkit for UNIX/Linux (X11), Microsoft Windows, and MacOS X. It supports 3D graphics via OpenGL.

filtered back-projection (FBP) Algorithm for the inversion of the two-dimensional Radon transform.

fluoroscopy Medical imaging modality for obtaining real-time x-ray images of a living patient. It is commonly used in IR procedures for the manipulation of guidewires.

four-dimensional (4D) Or four dimensions.

fragment Each fragment corresponds to a pixel into the framebuffer.

fragment program Microprogram substituting major parts of the computation of the traditional fixed function of the rasterization unit. A fragment program is used to compute the final colour of each fragment. A fragment program is performed once per fragment.

framebuffer object (FBO) Contains state information for the OpenGL framebuffer and its set of images.

frames per second (FPS) Number of images computed in a second.

functional magnetic resonance imaging (fMRI) MRI technique allowing the mapping of the function of regions of the brain.

G

gastrostomy Procedure that consists in opening permanently the external surface of the abdominal wall to the stomach.

gauge Needle calibres in medicine are measured in gauge.

general-purpose computing on graphics processing units (GPGPU) A trend in scientific computing to perform calculations on GPUs instead of the CPU.

gigaelectron volts (GeV) Unit of energy.

graphical user interface (GUI) A kind of user/computer interface in which the user interacts with a computer using graphical elements called "widgets".

graphics processing unit (GPU) A special hardware processor used in computer graphics.

H

haptic Of, or relating to, the sense of touch; tactile.

haptic interface point (HIP) single point corresponding to the haptic device's end effector.

Haptic Operative Realistic Ultrasonography Simulator (HORUS) Ultrasound guided needle puncture simulator develloped at IRCAD.

High Level Shader Language (HLSL) Also called High Level Shading Language. It is a proprietary shading language developed by Microsoft to provide more direct control of the graphics pipeline.

Hounsfield unit (HU) Pixel intensity unit used by most of the medical x-ray CT scanners. The standard Hounsfield scale is $[-1000, 1000]$.

HyperText Markup Language (HTML) Standard language for formatting documents on the World Wide Web. HTML is a simplified subset of Standard Generalized Markup Language (SGML).

hypoechoic Dark region in an ultrasound image in which the echoes are weaker or fewer than normal.

I

Insight Segmentation and Registration Toolkit (ITK) An open source API for image segmentation and registration.

Institut de Recherche contre les Cancers de l'Appareil Digestif (IRCAD) See `<http://www.ircad.fr/>` (accessed 11th October 2006).

Interventional Cardiology Training System (ICTS) .

interventional radiology (IR) A quite recent medical speciality introduced in 1964 which consists of minimally invasive diagnostics or treatments guided by images (usually fluoroscopy, ultrasound, CT, or MRI).

K

kiloelectron volts (keV) Unit of energy.

kilohertz (kHz) Unit of frequency equal to 1^3 hertz.

L

lesion Abnormal tissue usually damaged by trauma or disease.

look up table (LUT) data structure, usually an array, used to replace a runtime computation with a lookup operation.

lumen canal, duct, or cavity of a tubular organ.

M

magnetic resonance angiography (MRA) Angiography performed using magnetic resonance imaging (MRI).

magnetic resonance imaging (MRI) Imaging modality used in hospital to acquire 3D data sets from patients; formerly called Nuclear Magnetic Resonance Imaging (NMRI).

Manchester Visualization Centre (MVC) Research group at the University of Manchester.

mathematical morphology Theory for extracting image components that was initiated in the sixties by Georges Matheron and Jean Serra.

maximum intensity projection (MIP) Ray casting technique to pick out the voxel of maximum intensity.

megahertz (MHz) Unit of frequency equal to 1^6 hertz.

mesh Often called polygon mesh. Three-dimensional object described by a network of polygons, usually triangles.

million electron volts (MeV) Unit of energy also called megaelectron volt.

multiplanar reconstruction (MPR) Oblique image computed from the original set of parallel CT or MRI slices.

N

N.A. not applicable.

National Health Service (NHS) Is the public, national health service in Great Britain.

nephrostomy Establishment of an opening between a renal pelvis and the outside of the body.

non-destructive testing (NDT) Testing method which does not destroy the original object, such as x-ray computerized tomography.

nuclear magnetic resonance imaging (NMRI) Imaging modality used in hospital to acquire 3D data sets from patients; called nowadays Magnetic Resonance Imaging (MRI) due to the negative connotations associated with the term nuclear.

O

one-dimensional (1D) Or one dimension.

OpenGL® Shading Language (GLSL) Also called GLSLang. It is a platform independent high level shading language developed by the OpenGL architecture review board (OpenGL ARB) to provide more direct control of the graphics pipeline. This language is based on the C programming language.

OpenGL® Shading Language (GLSLang) see GLSL.

OpenGL Utility Library (GLU) Library of functions that supports quadrics, NURBS, complex polygons, matrix utilities, and more.

OpenGL Utility toolkit (glut) Library of functions developed by Mark Kilgard to hide the complexities of different windowing system APIs. Glut was originally used in Kilgard's book *OpenGL Programming for the X Window System* (1996) to illustrate OpenGL examples.

P

partial differential equation (PDE) A type of differential equation.

personal computer (PC) A computer suitable for personal usage (formerly called IBM PC compatible).

picture element (pixel) 2D discrete pictures are represented by arrays of pixels. A pixel represents a value on a two dimensional regular grid.

polyvinyl chloride (PVC) A common thermoplastic resin.

probe Ultrasound probe, see transducer.

proxy A small sphere representing the desired position of the haptic device. Although the device can penetrate into the object with which it is interacting, the proxy always remains outside of its surface.

R

Radiological Society of North America (RSNA) See <http://www.rsna.org/> (accessed 11th October 2006).

rasterization Simultaneous conversion of geometric and pixel data into fragments.

® registered trademark.

S

Seldinger technique medical procedure, developed by Sven-Ivar Seldinger in 1953 [232], to obtain access to blood vessels and other hollow organs using a sharp hollow needle, eventually guided by imaging.

sinogram The resulting image after a Radon transform. It is the measurements observed during the data acquisition of a computerized tomography. It is built from the set of x-ray projections at successive angles. Tomographic reconstruction consists in estimating the original object from this sinogram.

slab Region between a pair of parallel planes.

Society of Interventional Radiology (SIR) See <http://www.sirweb.org/> (accessed 11th October 2006).

stenting Surgical procedure for inserting a stent into a blood vessel.

T

texture element (texel) Textures are represented by arrays of texels. A texel represents the colour (or greyscale intensity) of the texture to apply on the corresponding fragment.

three-dimensional (3D) Or three dimensions.

thrombectomy Procedure that consists in removing a blood clot from a blood vessel.

time gain compensation (TGC) Increasing amplification of ultrasound echoes from increasing depths. It is used to compensate the progressive attenuation of deeper echoes.

timer A callback function to be triggered at regular interval of time (usually in milliseconds).

TM Trademark.

transducer Ultrasound transducers are also called ultrasound probe. They are devices used in medicine to generate, detect and measure directional acoustic fields (or beams).

transluminal Passing or occurring across a lumen.

Tuohy needle Hollow needle slightly curved at the end used for inserting epidural catheters.

two-dimensional (2D) Or two dimensions.

U

ultrasound (US) Sound with a frequency greater than 20 000 hertz, approximately the upper limit of human hearing.

V

vertebroplasty Procedure where bone cement is percutaneously injected into a fractured vertebra.

vertex program Microprogram substituting major parts of the computation of the traditional fixed function of the geometry processing unit. A vertex program is used to compute the transformation of the vertex, and allow arbitrary modifications of the vertex attributes. A vertex program is performed once per vertex.

virtual environment (VE) Collection of technologies that allow people to interact efficiently with 3D computerized databases in real-time using their natural senses and skills.

Virtual Reality Modeling Language (VRML) VRML 97 is the established ISO standard used on the Web to describe three-dimensional data.

Visualization ToolKit (VTK) An open source API for 3D computer graphics, image processing, and visualization. It consists of C++ class libraries, and several wrappers for Tcl, Java, and Python.

volume element (voxel) 3D discrete volume data sets are represented by arrays of voxels. A voxel represents a value on a three dimensional regular grid.

X

X3D X3D and its ancestor VRML are the established ISO standards used on the Web to describe three-dimensional data. X3D uses an XML syntax.

BIBLIOGRAPHY

[1] 3Dconnexion. Website. `<http://www.3dconnexion.com/>` (accessed 18th December 2006).

[2] S. Agostinelli, J. Allison, K. Amako, J. Apostolakis, H. Araujo, P. Arce, M. Asai, D. Axen, S. Banerjee, G. Barrand, and et al. GEANT4–a simulation toolkit. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 506(3):250–303, July 2003. DOI: 10.1016/S0168-9002(03)01368-8.

[3] D. Aiger and D. Cohen-Or. Real-time ultrasound imaging simulation. *Real-Time Imaging*, 4(4):263–274, Aug. 1998. 10.1006/rtim.1997.0089.

[4] A. L. Ames, D. R. Nadeau, and J. L. Moreland. *VRML 2.0 Sourcebook*. John Wiley & Sons Inc, 2nd edition, dec 1996. ISBN: 0-471-16507-7.

[5] Ascension Technology Corporation. Website. `<http://www.ascension-tech.com/>` (accessed 28th Jan 2008).

[6] R. Aster, B. Borchers, and C. H. Thurber. *Parameter Estimation and Inverse Problems*. Academic Press, 2004. ISBN: 0120656043.

[7] F. H. Attix and W. C. Roesch. *Radiation dosimetry*, chapter X-ray and γ-ray interactions. Academic press, 2nd edition, 1968.

[8] M. B. Aufderheide, D. M. Slone, and A. E. Schach von Wittenau. HADES, a radiographic simulation code. In D. O. Thompson and D. E. Chimenti, editors, *Review of Progress in Quantitative Nondestructive Evaluation*, volume 20, pages 507–513. American Institute of Physics, 2001.

[9] R. D. Badawi. Nuclear medicine. *Physics Education*, 36(6):452–459, Nov. 2001. DOI: 10.1088/0031-9120/36/6/302.

[10] D. H. Ballard and C. M. Brown. *Computer Vision*. Prentice Hall, 1982. ISBN: 0131653164.

[11] A. Ballato. Piezoelectricity: old effect, new thrusts. *IEEE Transactions on Ultrasonics, Ferroelectrics and Frequency Control*, 42(5):916–926, Sept. 1995. DOI: 10.1109/58.464826.

[12] R. Barkmann, W. Timm, S. Sakata, M. Heller, and C.-C. Gluer. Simulation of ultrasound interaction with bone structure based on 3D high-resolution-magnetic-resonance- and micro-CT-images. In *Proceedings of the 22nd Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, volume 4, pages 2666–2669, Chicago, IL, USA, July 2000. DOI: 10.1109/IEMBS.2000.901407.

[13] C. D. Barry, C. P. Allott, N. W. John, P. M. Mellor, P. A. Arundel, D. S. Thomson, and J. C. Waterton. Three-dimensional freehand ultrasound: Image reconstruction and volume analysis. *Ultrasound in medicine & biology*, 23(8):1209–1224, 1997. DOI: 10.1016/S0301-5629(97)00123-3.

[14] C. Basdogan, S. De, J. Kim, M. Muniyandi, H. Kim, and M. A. Srinivasan. Haptics in minimally invasive surgical simulation and training. *IEEE Computer Graphics & Applications*, 24(2):56–64, 2004. DOI: 10.1109/MCG.2004.10021.

[15] A. Benassarou, E. Bittar, N. W. John, and L. Lucas. MC slicing for volume rendering applications. In *Workshop on Computer Graphics and Geometric Modeling, CGGM 2005*, volume 3515 of *Lecture Notes in Computer Science*, pages 314–321. Springer Berlin / Heidelberg, Jan. 2005. DOI: 10.1007/11428848_39.

[16] G. Berci and K. Forde. History of endoscopy what lessons have we learned from the past? *Surgical Endoscopy*, 14(1):5–15, Jan. 2000. DOI: 10.1007/s004649900002.

[17] M. J. Berger, J. H. Hubbell, S. M. Seltzer, J. Chang, J. S. Coursey, R. Sukumar, , and D. S. Zucker. Xcom: Photon cross sections database. Online database, NIST, 2005. Accessible at: `<http://physics.nist.gov/PhysRefData/Xcom/Text/XCOM.html>` (accessed 6th October 2006).

[18] J. Blewett. R.R. Wilson prize lecture: Advantures with accelerators. In *Proceedings of the 1993 Particle Accelerator Conference*, volume 4, pages 2546–2550, Washington, DC, USA, May 1993. IEEE. DOI: 10.1109/PAC.1993.309384.

[19] E. M. Bocage. Technique and mechanism of a moving x-ray film. French patent No. 536464, French Patent Office, Paris, France, 1921. Quoted in [75].

[20] A. Bonin, B. Chalmond, and B. Lavayssière. Monte-Carlo simulation of industrial radiography images and experimental desings. *NDT & E International*, 35(8):503–510, Dec. 2002. DOI: 10.1016/S0963-8695(02)00014-2.

[21] U. D. Bordoloi and H.-W. Shen. Space efficient fast isosurface extraction for large datasets. In *Proceedings of the 14th IEEE Visualization 2003 (VIS '03)*, pages 201–208, Washington, DC, USA, 2003. IEEE Computer Society. DOI: 10.1109/VISUAL.2003.1250373.

[22] A. Bounaïm, S. Holm, W. Chen, and Å. Ødegård. Detectability of breast lesions with CARI ultrasonography using a bioacoustic computational approach. *Computers & Mathematics with Applications*, 54(1):96–106, 2007. DOI: 10.1016/j.camwa.2006.03.037.

[23] A. Bounaïm, S. Holm, W. Chen, Å. Ødegård, A. Tveito, and K. Thomenius. FETD simulation of wave propagation modeling the cari breast sonography. In *Computational Science and Its Applications - ICCSA 2003*, volume 2668/2003 of *Lecture Notes in Computer Science*, page 979, 2003.

[24] M. Brehmer and R. Swartz. Training on bench models improves dexterity in ureteroscopy. *European Urology*, 48(3):458–463, Sept. 2005. DOI: 10.1016/j.eururo.2005.04.031.

[25] M. Brehmer and D. Tolley. Validation of a bench model for endoscopic surgery in the upper urinary tract. *European Urology*, 42(2):175–180, Aug. 2002. DOI: 10.1016/S0302-2838(02)00265-8.

[26] P. N. Brett, T. J. Parker, A. J. Harrison, T. A. Thomas, and A. Carr. Simulation of resistance forces acting on surgical needles. *Proceedings of the I MECH E Part H Journal of Engineering in Medicine*, 211(4):335–347, Sept. 1997.

[27] M. Bridges and D. L. Diamon. The financial impact of teaching surgical residents in the operating room. *The American Journal of Surgery*, 177(1):28–32, Jan. 1999. DOI: 10.1016/S0002-9610(98)00289-X.

[28] K. Brodlie and J. Wood. Recent advances in visualization of volumetric data: State of theart report. In *Eurographics*, pages 65–84, Interlaken, Switzerland, 2000.

[29] K. Calman. Hospital doctors: training for the future. *British Journal of Obstetrics and Gynaecology*, 102(5):354–356, May 1995. DOI: 10.1111/j.1471-0528.1995.tb11283.x.

[30] G. G. Cameron and P. E. Undrill. Rendering volumetric medical image data on a SIMD architecture computer. In *Proceedings of the Third Eurographics Workshop on Rendering*, pages 135–145, Bristol, England, May 1992.

[31] H. Carr, J. Snoeyink, and U. Axen. Computing contour trees in all dimensions. *Computational Geometry*, 24(2):75–94, Feb. 2003. DOI: 10.1016/S0925-7721(02)00093-7.

[32] N. A. Carr, J. D. Hall, and J. C. Hart. The ray engine. In *HWWS '02: Proceedings of the ACM SIGGRAPH/EUROGRAPHICS conference on Graphics hardware*, pages 37–46, Aire-la-Ville, Switzerland, Switzerland, 2002. Eurographics Association.

[33] J. A. Carrino and F. A. Jolesz. MRI-guided interventions. *Academic Radiology*, 12(9):1063–1064, Sept. 2005. DOI: 10.1016/j.acra.2005.06.008.

[34] E. E. Catmull. *A subdivision algorithm for computer display of curved surfaces*. PhD thesis, Department of Computer Science, University of Utah, Salt Lake City, 1974.

[35] G. A. Chapman, D. Johnson, and A. R. Bodenham. Visualisation of needle position using ultrasonography. *Anaesthesia*, 61(2):148–158, 2006. DOI: 10.1111/j.1365-2044.2005.04475.x.

[36] R. C. Chivers. The scattering of ultrasound by human tissues–some theoretical models. *Ultrasound in medicine & biology*, 3(1):1–13, 1977.

[37] H. S. Choi, D. R. Haynor, and Y. Kim. Partial volume tissue classification of multichannel magnetic resonance images-a mixel model. *IEEE Transactions on Medical Imaging*, 10(3), Sept. 1991.

[38] P. Clœtens, W. Ludwig, E. Boller, F. Peyrin, M. Schenker, and J. Baruchel. 3d imaging using coherent synchrotron radiation. *Image Analysis & Stereology*, 21(Suppl 1):S75–S85, Dec. 2002.

[39] B. A. Coghlan and M. G. Taylor. Directional doppler techniques for detection of blood velocities. *Ultrasound in medicine & biology*, 2(3):181–188, June 1976.

[40] F. Conti, F. Barbagli, D. Morris, and C. Sewell. CHAI: An open-source library for the rapid development of haptic scenes. Demo paper presented at IEEE World Haptics, Pisa, Italy, Mar. 2005.

[41] CR*a*IVE - Collaboration in Radiological Interventional Virtual Environments. Website. `<http://www.craive.org.uk/>` (accessed 24th June 2007).

[42] T. J. Cullip and U. Neumann. Accelerating volume reconstruction with 3D texture hardware. Technical Report TR93-027, Department of Computer Science at the University of North Carolina, Chapel Hill, NC, USA, 1993.

[43] i. CyberLogic. Website. `<http://www.cyberlogic.org/>` (accessed 28th October 2007).

[44] J. Dankelman, M. Wentink, C. A. Grimbergen, H. G. Stassen, and J. Reekers. Does virtual reality training make sense in interventional radiology? training skill-, rule- and knowledge-based behavior. *Cardiovascular and Interventional Radiology*, 27(5):417–421, 2004. DOI: 10.1007/s00270-004-0250-y.

[45] S. L. Dawson, S. Cotin, D. Meglan, D. W. Shaffer, and M. A. Ferrell. Designing a computer-based simulator for interventional cardiology training. *Catheterization and Cardiovascular Interventions*, 51(4):522–527, Dec. 2000. DOI: 10.1002/1522-726X(200012)51:4¡522::AID-CCD30¿3.0.CO;2-7.

[46] D. L. Dekker, R. L. Piziali, and E. Dong Jr. A system for ultrasonically imaging the human heart in threedimensions. *Computers and biomedical research*, 7:544–553, Dec. 1974.

[47] Department of Trade and Industry Working. Working time regulations. Available at `<http://www.dti.gov.uk/employment/employment-legislation/working-time-regs/>` (accessed 24th June 2007).

[48] S. P. DiMaio and S. E. Salcudean. Needle insertion modelling for the interactive simulation of percutaneous procedures. In T. Dohi and R. Kikinis, editors, *Medical Image Computing and Computer-Assisted Intervention - MICCAI 2002: 5th International Conference, Proceedings, Part II*, volume 2489 of *Lecture Notes in Computer Science*, pages 253–260, Tokyo, Japan, Sept. 2002. Springer-Verlag.

[49] S. P. DiMaio and S. E. Salcudean. Interactive simulation of needle insertion models. *IEEE Transactions on Biomedical Engineering*, 52(7):1167–1179, July 2005. DOI: 10.1109/TBME.2005.847548.

[50] A. Doi and A. Koide. An efficient method for visualization of 3-dimensional functions by triangulating equi-valued surfaces. *IEICE Transactions*, E74(1):214–224, Jan. 1991.

[51] R. F. Dondelinger, M. P. Ghysels, D. Brisbois, E. Donkers, F. R. Snaps, J. Saunders, and J. Devière. Relevant radiological anatomy of the pig as a training model in interventional radiology. *European Radiology*, 8(7):1254–1273, Sept. 1998. DOI: 10.1007/s003300050545.

[52] C. T. Dotter and M. P. Judkins. Transluminal treatment of arteriosclerotic obstruction, description of a new technic and a preliminary report of its application. *Circulation*, 30:654–670, Nov. 1964.

[53] R. A. Drebin, L. Carpenter, and P. Hanrahan. Volume rendering. In *SIGGRAPH '88: Proceedings of the 15th annual conference on Computer graphics and interactive techniques*, pages 65–74, New York, NY, USA, 1988. ACM Press. DOI: 10.1145/54852.378484.

[54] P. Duvauchelle, N. Freud, V. Kaftandjian, and D. Babot. A computer code to simulate x-ray imaging techniques. *Nuclear Instruments and Methods in Physics Research B*, 170(1-2):245–258, sep 2000. DOI: 10.1016/S0168-583X(00)00185-3.

[55] Education Board of the Faculty of Clinical Radiology, The Royal College of Radiologists, London, UK. *Structured Training in Clinical Radiology*, 4th edition, Sept. 2004.

[56] H. Elleaume, S. Fiedler, F. Estève, B. Bertrand, A. M. Charvet, P. Berkvens, G. Berruyer, T. Brochard, G. L. Duc, C. Nemoz, M. Renier, P. Suortti, W. Thomlinson, and J. F. L. Bas. First human transvenous coronary angiography at the european synchrotron radiation facility. *Physics in Medicine and Biology*, 45(9):L39–L43, Sept. 2000. DOI: 10.1088/0031-9155/45/9/102.

[57] M. Ellis. DELFI ultrasound simulation tool. `<http://www.mathworks.com/matlabcentral/fileexchange/loadFile.do?objectId=13970&objectType=FILE>` (accessed 28th October 2007).

[58] M. A. Ellis, D. Guenther, and W. F. Walker. A spline-based approach for computing spatial impulse responses. *IEEE Transactions on Ultrasonics, Ferroelectrics and Frequency Control*, 54(5):1045–1054, May 2007. DOI: 10.1109/TUFFC.2007.350.

[59] M. A. Ellis and W. F. Walker. P2f-2 piecewise-linear approximation for improved accuracy in near-field ultrasound simulation. In *IEEE Ultrasonics Symposium 2006*, pages 1698–1701, Oct. 2006. DOI: 10.1109/ULTSYM.2006.428.

[60] K. Engel, M. Hadwiger, J. M. Kniss, A. E. Lefohn, C. R. Salama, and D. Weiskopf. Real-time volume graphics. In *SIGGRAPH '04: ACM SIGGRAPH 2004 Course Notes*, page 29, New York, NY, USA, 2004. ACM Press. DOI: 10.1145/1103900.1103929.

[61] K. Engel, M. Hadwiger, J. M. Kniss, C. Rezk-Salama, and D. Weiskopf. *Real-time volume graphics*. A. K. Peters, 2006.

[62] A. X. Falcao, J. K. Udupa, S. Samarasekera, S. Sharma, B. E. Hirsch, and R. de Alencar Lofufo. User-steered image segmentation paradigms: Live-wire and live-lane. *Graphical Models and Image Processing*, 60(4):223–260, July 1998. DOI: 10.1006/gmip.1998.0475.

[63] R. Fernando, editor. *GPU Gems: Programming Techniques, Tips & Tricks for Real-Time Graphics*. Addison-Wesley, 2004. ISBN: 0321228324.

[64] R. Fernando and M. J. Kilgard. *The Cg Tutorial: The Definitive Guide to Programmable Real-Time*. Addison-Wesley Professional, Feb. 2003. ISBN: 0-321-19496-9.

[65] G. R. Ferretti, I. Bricault, and M. Coulomb. Virtual tools for imaging of the thorax. *European Respiratory Journal*, 18:381–392, 2001.

[66] Fltk. <http://www.fltk.org/> (accessed 26th October 2007).

[67] J. D. Foley, A. van Dam, S. K. Feiner, and J. F. H. s. *Computer Graphics: Principles and Practice in C*. Addison Wesley Professional, 2nd edition, 1996. ISBN: 0-201-84840-6.

[68] Force Dimension. Website. <http://www.forcedimension.com/> (accessed 9th August 2006).

[69] C. Forest, O. Comas, C. Vaysière, L. Soler, and J. Marescaux. Ultrasound and needle insertion simulators built on real patient-based data. In J. D. Westwood, R. S. Haluck, H. M. Hoffman, G. T. M. R. Phillips, R. A. Robb, and K. G. Vosburgh, editors, *Medicine*

Meets Virtual Reality 15, volume 125 of *Studies in Health Technology and Informatics*, pages 136–139. IOS Press, 2007.

[70] B. D. Fornage. A simple phantom for training in ultrasound-guided needle biopsy using the freehand technique. *Journal of Ultrasound in Medicine*, 8(12):701–703, Dec. 1989.

[71] A. Franke and H. P. Kühl. Second-generation real-time 3D echocardiography: a revolutionary new technology. *Medica Mundi*, 47(2):34–40, Aug. 2003.

[72] N. Freud. *Modélisation et simulation de système d'imagerie par rayons X et gamma*. PhD thesis, INSA-Lyon Scientific and Technical University, France, CNDRI, Laboratory of Nondestructive Testing Using Ionizing Radiations, 2003.

[73] N. Freud, P. Duvauchelle, J. M. Létang, and D. Babot. Fast and robust ray casting algorithms for virtual X-ray imaging. *Nuclear Instruments and Methods in Physics Research Section B: Beam Interactions with Materials and Atoms*, 248(1):175–180, July 2006. DOI: 10.1016/j.nimb.2006.03.009.

[74] N. Freud, J. M. Létang, C. Mary, C. Boudou, C. Ferrero, H. Elleaume, A. Bravin, F. Estève, and D. Babot. Fast dose calculation for stereotactic synchrotron radiotherapy. In *29th IEEE EMBS Annual International Conference*, 2007.

[75] G. W. Friedland and B. O. Thurber. The birth of CT. *American Journal of Roentgenology*, 167:1365–1370, 1996.

[76] O. Gerovichev, P. Marayong, and A. M. Okamura. The effect of visual and haptic feedback on manual and teleoperated needle insertion. In *MICCAI '02: Proceedings of the 5th International Conference on Medical Image Computing and Computer-Assisted Intervention - Part I*, volume 2488/2002 of *Lecture Notes in Computer Science*, pages 147–154. Springer-Verlag, Sept. 2002. DOI: 10.1007/3-540-45786-0_19.

[77] O. Gerovichev, P. Marayong, and A. M. Okamura. The effect of visual and haptic feedback on computer-assisted needle insertion. *Computer Aided Surgery*, 9(6):243–249, 2004. DOI: 10.1080/10929080500190441.

[78] S. F. Gibson. 3D chainmail: a fast algorithm for deforming volumetric objects. In *Proceedings of the 1997 symposium on Interactive 3D graphics*, pages 149–154, Providence, Rhode Island, United States, 1997. ACM Press.

[79] E. Gobbetti, M. Tuveri, G. Zanetti, and A. Zorcolo. Catheter insertion simulation with co-registered direct volume rendering and haptic feedback. In J. D. Westwood, H. M. Hoffman, G. T. Mogel, D. Stredney, and R. A. Robb, editors, *Medicine Meets Virtual Reality 2000*, volume 70 of *Studies in Health Technology and Informatics*, pages 96–98. IOS Press, Jan. 2000.

[80] F. Goetz, T. Junklewitz, and G. Domik. Real-time marching cubes on the vertex shader. In *Short Presentations of Eurographics 2005*, 2005.

[81] B. B. Goldberg. Obstetric US imaging: The past 40 years. *Radiology*, 215(3):622–629, 2000.

[82] R. C. Gonzalez and R. E. Woods. *Digital Image Processing*. Prentice Hall, Inc., 2nd edition, 2002. ISBN: 0-201-18075-8.

[83] D. A. Gould, J. A. Reekers, D. O. Kessel, N. C. Chalmers, M. Sapoval, A. A. Patel, G. J. Becker, M. J. Lee, and L. Stockx. Simulation devices in interventional radiology: Caveat emptor. *CardioVascular and Interventional Radiology*, 29(1):4–6, Feb. 2006. DOI: 10.1007/s00270-005-0354-z.

[84] H. Gouraud. Continuous shading of curved surfaces. *IEEE Transactions on Computers*, 20(6):623–628, 1971.

[85] General-purpose computation using graphics hardware. Website. `<http://www.gpgpu.org/>` (accessed 24th October 2007).

[86] A. Grama, A. Gupta, G. Karypis, and V. Kumar. *Introduction to Parallel Computing*. Pearson Education Limited, 2nd edition edition, 2003. ISBN: 0-201-64865-2.

[87] S. Grimm, S. Bruckner, A. Kanitsar, and E. Groller. Memory efficient acceleration structures and techniques for CPU-based volume raycasting of large data. In *IEEE Symposium on Volume Visualization and Graphics*, pages 1–8, 2004. DOI: 10.1109/SVVG.2004.8.

[88] A. Gueziec and R. Hummel. Exploiting triangulated surface extraction using tetrahedraldecomposition. *IEEE Transactions on Visualization and Computer Graphics*, 1(4):328–342, Dec. 1995. DOI: 10.1109/2945.485620.

[89] C. D. Hansen and C. R. Johnson. *Visualization Handbook*. Academic Press, 2004. ISBN: 012387582X.

[90] Haptik. Website. <http://www.haptiklibrary.org/> (accessed 15th December 2006).

[91] M. Harris. Physically-based simulation on gpus. SIGGRAPH 2004 GPGPU COURSE, 2004. Available at http://www.gpgpu.org/s2004/slides/harris. SimulationGPU.ppt.

[92] M. J. Harris, W. V. Baxter, T. Scheuermann, and A. Lastra. Simulation of cloud dynamics on graphics hardware. In *HWWS '03: Proceedings of the ACM SIGGRAPH/EUROGRAPHICS conference on Graphics hardware*, pages 92–101, Aire-la-Ville, Switzerland, Switzerland, 2003. Eurographics Association.

[93] A. E. Healey, J. C. Evans, M. G. Murphy, S. Powell, T. V. How, D. Groves, F. Hatfield, B. M. Diaz, and D. A. Gould. In vivo force during arterial interventional radiology needlepuncture procedures. In J. D. Westwood, R. S. Haluck, H. M. Hoffman, G. T. Mogel, R. Phillips, R. A. Robb, and K. G. Vosburgh, editors, *Medicine Meets Virtual Reality 13*, volume 111 of *Studies in Health Technology and Informatics*, pages 178–184. IOS Press, 2005.

[94] E. A. M. Heijnsdijk, A. Pasdeloup, A. J. van der Pijl, J. Dankelman, and D. J. Gouma. The influence of force feedback and visual feedback in grasping tissue laparoscopically. *Surgical Endoscopy*, 18(6):980–985, June 2004.

[95] K. Held, E. Rota Kops, B. J. Krause, W. M. r. Wells, R. Kikinis, and H. W. Müller-Gärtner. Markov random field segmentation of brain MR images. *IEEE Transactions on Medical Imaging*, 16(6):878–886, Dec. 1997.

[96] G. T. Herman and H. K. Liu. Display of three-dimensional information in computed tomography. *Journal of Computer Assisted Tomography*, 1(1):155–160, Jan. 1977.

[97] G. Hiebert. *OpenAL Programmer's Guide: OpenAL Versions 1.0 and 1.1.* Creative Technology Limited, revision 1.2 edition, Dec. 2006. Published online at <ftp://216.61.164.51/OpenAL_Programmers_Guide.pdf> (accessed 18th December 2006).

[98] C. R. Hill, J. C. Bamber, and G. R. ter Haar, editors. *Physical Principles of Medical Ultrasonics*, chapter Basic Acoustic Theory, pages 1–40. Wiley, 2004. ISBN: 0-471-97002-6.

[99] K. H. Hoehne, R. L. Delapaz, R. Bernstein, and R. C. Taylor. Combined surface display and reformatting for the three-dimensional analysis of tomographic data. *Investigative Radiology*, 22:658–, 1987.

[100] J. H. Holmes, D. H. Howry, G. J. Posakony, and C. R. Cushman. The ultrasonic visualization of soft tissue structures in the human body. *Transactions of the American Clinical and Climatological Association*, 66:208–225, 1954.

[101] M. Hopf and T. Ertl. Hardware based wavelet transformations. In *Workshop on Vision, Modeling, and Visualization 1999*, pages 317–328, Nov. 1999.

[102] A. Hostettler, C. Forest, A. Forgione, L. Soler, and J. Marescaux. A real-time ultrasonography simulator based on CT-scan images. In J. D. Westwood, R. S. Haluck, H. M. Hoffman, G. T. Mogel, R. Phillips, R. A. Robb, and K. G. Vosburgh, editors, *Medicine Meets Virtual Reality 13*, volume 111 of *Studies in Health Technology and Informatics*, pages 191–193. IOS Press, Jan. 2005.

[103] G. N. Hounsfield. Computerized transverse axial scanning (tomography): Part 1. description of system. *The British journal of radiology*, 46:1016–1022, 1973.

[104] P. Howry and W. Bliss. Ultrasonic visualization of soft tissue structures of the body. *Journal of laboratory and clinical medicine*, 40(4):579–592, Oct. 1952.

[105] J. Huang, K. Mueller, N. Shareef, and R. Crawfis. FastSpats: Optimized splatting on rectilinear grids. In *Proceedings of IEEE Visualization 2000*, pages 219–226, 2000. DOI: 10.1109/VISUAL.2000.885698.

[106] J. W. Hunt, M. Arditi, and F. F. S. Ultrasound transducers for pulse-echo medical imaging. *IEEE transactions on bio-medical engineering*, 30(8):453–481, Aug. 1983.

[107] M. A. Hurn, K. V. Mardia, T. J. Hainsworth, J. Kirkbride, and E. Berry. Bayesian fused classification of medical images. *IEEE Transaction on Medical Imaging*, 15(6):850–858, 1996. DOI: 10.1109/42.544502.

[108] IEEE UFFC. Ultrasonics software. `<http://www.ieee-uffc.org/ulmain.asp?view=software>` (accessed 29th October 2007).

[109] K. E. H. III", T. Culver, J. Keyser, M. Lin, and D. Manocha. Fast computation of generalized voronoi diagrams using graphics hardware. In *SCG '00: Proceedings of the sixteenth annual symposium on Computational geometry*, pages 375–376, New York, NY, USA, 2000. ACM Press.

[110] Immersion Medical. Website. `<http://www.immersion.com/medical/>` (accessed 22nd March 2007).

[111] F. Inanc, J. N. Gray, T. Jensen, and J. Xu. Human body radiography simulations: development of a virtual radiography environment. In *Physics of Medical Imaging*, volume 3336, pages 830–837, San Diego, California, February 1998. SPIE – The International Society for Optical Engineering.

[112] Incomes Data Services. European working time directive - IDS working time. Available at `<http://www.incomesdata.co.uk/information/worktimedirective.htm>` (accessed 24th June 2007).

[113] Insight segmentation and registration toolkit (ITK). Webpage. `<http://www.itk.org/>` (accessed 31st January 2008).

[114] B. Itkowitz, J. Handley, and W. Zhu. The OpenHaptics^{TM} toolkit: a library for adding 3D Touch^{TM} navigation and haptics to graphics applications. In *First Joint Eurohaptics Conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems 2005 (WHC '05)*, pages 590–591. IEEE Computer Society, Mar. 2005. DOI: 10.1109/WHC.2005.133.

[115] G. James. *Operations for Hardware-Accelerated Procedural Texture Animation*, volume 2 of *Game Programming Gems*, pages 497–500. Charles River Media, 2001.

[116] F. Jensen. 100 years of X-rays. *Medica Mundi*, 40(3):109–120, 1995.

[117] J. A. Jensen. Field: A program for simulating ultrasound systems. In *Proceedings of the 10th Nordic-Baltic Conference on Biomedical Imaging*, volume 34(Supplement 1, Part 1) of *Medical & Biological Engineering & Computing*, pages 351–353, 1996.

[118] J. A. Jensen and N. B. Svendsen. Calculation of pressure fields from arbitrarily shaped, apodized, and excited ultrasound transducers. *IEEE Transactions on Ultrasonics, Ferroelectrics and Frequency Control*, 39(2):262–267, Mar. 1992. DOI: 10.1109/58.139123.

[119] G. Johansson and H. Carr. Accelerating marching cubes with graphics hardware. In *CASCON '06: Proceedings of the 2006 conference of the Center for Advanced Studies on Collaborative research*, page 39, New York, NY, USA, 2006. ACM Press. DOI: 10.1145/1188966.1189018.

[120] N. W. John and R. F. McCloy. Navigating and visualising three-dimensional data sets. *British Journal Radiology*, 77, Special Issue: Image Processing and Visualisation(Suppl 2):S108–S113, 2004. DOI: 10.1259/bjr/45222871.

[121] S. Johnson. Biopsy CT guided TA. Available at <http://www.hpv.cs.bangor.ac.uk/CRaIVE/CT%20guided%20biopsy%20task%20analysis.pdf> (accessed 24th June 2007).

[122] S. Johnson. Biopsy ultrasound guided TA. Available at <http://www.hpv.cs.bangor.ac.uk/CRaIVE/ultrasound%20guided%20biopsy%20task%20analysis.pdf> (accessed 24th June 2007).

[123] S. J. Johnson, A. E. Healey, J. C. Evans, M. G. Murphy, M. Crawshaw, and D. Gould. Physical and cognitive task analysis in interventional radiology. *Journal of Clinical Radiology*, 61(1):97–103, Jan. 2006. DOI: 10.1016/j.crad.2005.09.003.

[124] A. C. Kak and M. Slaney. *Principles of computerized tomographic imaging*. Society of Industrial and Applied Mathematics, 2001. ISBN: 089871494X. Electronic copy available online at <http://cobweb.ecn.purdue.edu/~malcolm/pct/pct-toc.html% guilsinglright (accessed 15 Feb 2007).

[125] A. Kanitsar, R. Wegenkittl, D. Fleischmann, and M. E. Groller. Advanced curved planar reformation: Flattening of vascular structures. In *Proceedings of the 14th IEEE Visualization 2003 (VIS '03)*, pages 43–50, Washington, DC, USA, 2003. IEEE Computer Society. DOI: 10.1109/VISUAL.2003.1250353.

[126] T. L. Kay and J. T. Kajiya. Ray tracing complex scenes. In *SIGGRAPH '86: Proceedings of the 13th annual conference on Computer graphics and interactive techniques*, pages 269–278, New York, NY, USA, 1986. ACM Press. DOI: 10.1145/15922.15916.

[127] G. Kedem and Y. Ishihara. Brute force attack on UNIX passwords with SIMD computer. In *SSYM'99: Proceedings of the 8th conference on USENIX Security Symposium*, pages 8–8, Berkeley, CA, USA, 1999. USENIX Association.

[128] S. F. Keevil. Magnetic resonance imaging in medicine. *Physics Education*, 36(6):476–485, Nov. 2001. DOI: 10.1088/0031-9120/36/6/305.

[129] Khronos Group. COLLADA - 3D asset exchange schema. Website. <http://www.khronos.org/collada/> (accessed 24th October 2007).

[130] G. Kindlmann, R. Whitaker, T. Tasdizen, and T. Moller. Curvature-based transfer functions for direct volume rendering: methods and applications. In *Proceedings of the 14th*

IEEE Visualization 2003 (VIS'03), pages 513–520, Washington, DC, USA, 2003. IEEE Computer Society. DOI: 10.1109/VISUAL.2003.1250414.

[131] Kitware. Paraview. `<http://www.paraview.org/>` (accessed 30th July 2007).

[132] Kitware. VolView. `<http://www.kitware.com/products/volview.html>` (accessed 30th July 2007).

[133] Kitware, Inc. The visualization toolkit (VTK). Webpage. `<http://www.vtk.org/>` (accessed 31st January 2008).

[134] Kitware Inc. Writing a metaimage header for raw data. Webpage. `<http://www.itk.org/HTML/Data.htm>` (accessed 18th August 2007).

[135] R. M. Koch, M. H. Gross, F. R. Carls, D. F. von Büren, G. Fankhauser, and Y. I. H. Parish. Simulating facial surgery using finite element models. In *SIGGRAPH '96: Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 421–428, New York, NY, USA, 1996. ACM Press.

[136] B. J. Krenning, M. M. Voormolen, and J. R. T. C. Roelandt. Assessment of left ventricular function by three-dimensional echocardiography. *Cardiovascular Ultrasound*, 1(1):12, Sept. 2003. DOI: doi:10.1186/1476-7120-1-12.

[137] J. Krüger and R. Westermann. Acceleration techniques for GPU-based volume rendering. In *Proceedings of the 14th IEEE Visualization 2003 (VIS'03)*, pages 287–292, Washington, DC, USA, Oct. 2003. IEEE Computer Society. DOI: 10.1109/VIS.2003.10001.

[138] Kyotokagaku co.,ltd. Website: Central venous catheter insertion simulator with ultrasound. `<http://www.kyotokagaku.com/products_medi_cvc.html>` (accessed 6th July 2007).

[139] P. Lacroute and M. Levoy. Fast volume rendering using a shear-warp factorization of the viewing transformation. In *SIGGRAPH '94: Proceedings of the 21st annual conference on Computer graphics and interactive techniques*, pages 451–458, New York, NY, USA, 1994. ACM Press. DOI: 10.1145/192161.192283.

[140] Laerdal Medical. Website. `<http://www.laerdal.com/>` (accessed 24th September 2006).

[141] D. Laney, S. P. Callahan, N. Max, C. T. Silva, S. Langer, and R. Frank. Hardware-accelerated simulated radiography. In *IEEE Visualization 2005 (VIS' 05)*, pages 343–350, 2005.

[142] S. D. Laycock and A. M. Day. Recent developments and applications of haptic devices. *Computer Graphics Forum*, 22(2):117–132, June 2003. DOI: 10.1111/1467-8659.00654.

[143] S. D. Laycock and A. M. Day. A survey of haptic rendering techniques. *Computer Graphics Forum*, 26(1):50–65, 2007. DOI: 10.1111/j.1467-8659.2007.00945.x.

[144] D. Lazos, Z. Kolitsi, and N. Pallikarakis. A software data generator for radiographic imaging investigations. *IEEE Transactions on Information Technology in Biomedicine*, 4(1):76–79, 2000.

[145] Y. Lee, D. Terzopoulos, and K. Waters. Realistic face modelling for animation. In *Proceedings of ACM SIGGRAPH' 95*, pages 55–62, 1995.

[146] A. E. Lefohn, J. M. Kniss, C. D. Hansen, and R. T. Whitaker. A streaming narrow-band algorithm: interactive computation and visualization of level sets. *IEEE Transactions on Visualization and Computer Graphics*, pages 422–433, July 2004. DOI: 10.1109/TVCG.2004.2.

[147] J. Lengyel, M. Reichert, B. R. Donald, and D. P. Greenberg. Real-time robot motion planning using rasterizing computer graphics hardware. In *SIGGRAPH '90: Proceedings of the 17th annual conference on Computer graphics and interactive techniques*, pages 327–335, New York, NY, USA, 1990. ACM Press.

[148] J.-M. Létang, N. Freud, and G. Peix. Signal-to-noise ratio criterion for the optimization of dual-energy acquisition using virtual X-ray imaging: application to glass wool. *Journal of Electronic Imaging*, 13(3):436–449, 2004.

[149] M. Levoy. Display of surfaces from volume data. *IEEE Computer Graphics and Applications*, 8(3):29–37, May 1988. DOI: 10.1109/38.511.

[150] M. Levoy. Efficient ray tracing of volume data. *ACM Transactions on Graphics*, 9(3):245–261, July 1990. DOI: 10.1145/78964.78965.

[151] N. Li, S.-H. Kim, J.-H. Suh, S.-H. Cho, J.-G. Choi, and M.-H. Kim. Virtual x-ray imaging techniques in an immersive casting simulation environment. *Nuclear Instruments and Methods in Physics Research B*, 262:143?–152, 2007.

[152] Y. Li and K. Brodlie. Soft object modelling with generalised chainmail - extending the boundaries of web-based graphics. *Computer Graphics Forum*, 22(4):717–727, 2003.

[153] Limbs and Things. Medical simulation models. Website. Website: <http://www.golimbs.com/products/products.php?sectid=67% guilsinglright (accessed 24th June 2007).

[154] G. S. Litynski. Endoscopic surgery: The history, the pioneers. *World Journal of Surgery*, 23(8):745–753, Aug. 1999. DOI: 10.1007/s002689900576.

[155] A. Lopes and K. Brodlie. Improving the robustness and accuracy of the marching cubes algorithm for isosurfacing. *IEEE Transactions on Visualization and Computer Graphics*, 9:16–29, 2003. DOI: 10.1109/TVCG.2003.1175094.

[156] W. E. Lorensen and H. E. Cline. Marching cubes: A high resolution 3D surface construction algorithm. In *SIGGRAPH '87: Proceedings of the 14th annual conference on Computer graphics and interactive techniques*, pages 163–169, New York, NY, USA, 1987. ACM Press. DOI: 10.1145/37401.37422.

[157] A. Lunderquist, K. Ivancev, S. Wallace, I. Enge, F. Laerum, and A. N. Kolbenstvedt. The acquisition of skills in interventional radiology by supervised training on animal models: a three-year multicenter experience. *CardioVascular and Interventional Radiology*, 18(4):209–211, July 1995. DOI: 10.1007/BF00239413.

[158] J. Marescaux, J. Clément, V. Tassetti, C. Koehl, S. Cotin, Y. Russier, D. Mutter, H. Delingette, and N. Ayache. Virtual reality applied to hepatic surgery simulation: the next revolution. *Annals of Surgery*, 228(5):627–634, Nov. 1998. DOI: 10.1097/00000658-199811000-00001.

[159] T. Massie and J. K. Salisbury. The PHANToM haptic interface, a device of probing virtual objects. In *Proceedings of the ASME Winter Annual Meeting, Symposium on Haptic Interfaces for Virtual Environments and Teleoperator system*, 1994.

[160] T. A. Matalon and B. Silver. US guidance of interventional procedures. *Radiology*, 174(1):43–47, Jan. 1990.

[161] R. McCloy and R. Stone. Science, medicine, and the future: Virtual reality in surgery. *British Medical Journal*, 323(7318):912–915, Oct. 2001. DOI: 10.1136/bmj.323.7318.912.

[162] T. McInerney and D. Terzopoulos. Deformable models in medical image analysis: A survey. *Medical Image Analysis*, 1(2):91–108, 1996.

[163] M. P. McNamara Jr and M. E. McNamara. Preparation of a homemade ultrasound biopsy phantom. *Journal of Clinical Ultrasound*, 17(6):456–458, 1989.

[164] W. A. McNeely, K. D. Puterbaugh, and J. J. Troy. Six degree-of-freedom haptic rendering using voxel sampling. In *SIGGRAPH '99: Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 401–408, New York, NY, USA, 1999. ACM Press/Addison-Wesley Publishing Co. 10.1145/311535.311600.

[165] T. McReynolds and D. Blythe. *Advanced Graphics Programming Using OpenGL*. The Morgan Kaufmann Series in Computer Graphics and Geometric Modeling. Morgan Kaufmann, 2005. ISBN: 1-55860-659-9.

[166] Mentice. Website. `<http://www.mentice.com/>` (accessed 9th August 2006).

[167] J. Merald. Interventional imaging: a new speciality in between radiology and surgery. *Bulletin de l'Académie Nationale de Médecine*, 175(7):1079–1084, Oct. 1991.

[168] G. Michael. X-ray computed tomography. *Physics Education*, 36(6):442–451, 2001. DOI: 10.1088/0031-9120/36/6/301.

[169] Microsoft Corporation. DirectX Developer Center. `<http://msdn.microsoft.com/directx/>` (accessed 13th December 2006).

[170] Mimic Technologies, Inc. Mantis workstation. `<http://www.mimic.ws/mantis.html>` (accessed 20th July 2007).

[171] B. Mora and D. S. Ebert. Low-complexity maximum intensity projection. *ACM Transactions on Graphics*, 24(4):1392–1416, 2005. DOI: 10.1145/1095878.1095886.

[172] H. B. Morgenbesser. Force shading for haptic shape perception in haptic virtual environments. Master's thesis, MIT, 1995.

[173] G. J. Morris Stiff, D. Clarke, J. Torkington, D. J. Bowrey, and R. E. Mansel. Training in the Calman era: what consultants say. *Bulletin of The Royal College of Surgeons of England*, 84(10):345–347, Nov. 2002.

[174] L. Mroz, H. Hauser, and E. Gröller. Interactive high-quality maximum intensity projection. *Computer Graphics Forum*, 19(3):341?350, Aug. 2000. DOI: 10.1111/1467-8659.00426.

[175] M. Muniyandi, S. Cotin, M. Srinivasan, and S. Dawson. Real-time PC based x-ray simulation for interventional radiology training. In J. D. Westwood, H. M. Hoffman, G. T. Mogel, R. Phillips, R. A. Robb, and D. Stredney, editors, *Medicine Meets Virtual Reality Conference (MMVR 11)*, volume 94 of *Studies in Health Technology and Informatics*, pages 233–239. IOSPress, 2003.

[176] National Electrical Manufacturers Association. Digital imaging and communications in medicine (dicom) part 3: Information object definitions. Website. <http://www.nema.org/stds/ps3-3.cfm> (accessed 24th October 2007).

[177] National Library of Medicine. The Visible Human Project. Website. <http://www.nlm.nih.gov/research/visible/> (accessed 27th October 2007).

[178] T. S. Newman and H. Yi. A survey of the marching cubes algorithm. *Computers & Graphics*, 30(5):854–879, Oct. 2006. DOI: 10.1016/j.cag.2006.07.021.

[179] Novint Technologies. Website. <http://www.novint.com/> (accessed 24th September 2006).

[180] Octaga. Website. <http://www.octaga.com/> (accessed 9th October 2006).

[181] A. M. Okamura, C. Simone, and M. D. O'Leary. Force modeling for needle insertion into soft tissue. *IEEE Transaction on Biomedical Engineering*, 51(10):1707–1716, Oct. 2004. DOI: 10.1109/TBME.2004.831542.

[182] OpenGL Architecture Review Board, D. Shreiner, M. Woo, J. Neider, and T. Davis. *OpenGL Programming Guide: The Official Guide to Learning OpenGL, Version 2*. Addison-Wesley Professional, 5th edition, Aug. 2005. ISBN: 0-321-33573-2.

[183] OpenSG Forum. OpenSG. <http://www.opensg.org/> (accessed 13th December 2006).

[184] OSG Community. OpenSceneGraph. <http://www.openscenegraph.org/> (accessed 13th December 2006).

[185] S. Osher and J. A. Sethian. Fronts propagating with curvature-dependent speed: algorithms based on hamilton-jacobi formulations. *J. Comput. Phys.*, 79(1):12–49, 1988.

[186] R. Otto and P. Deyhle. Guided puncture under real-time sonographic control. *Radiology*, 134(3):784–785, Mar. 1980.

[187] J. Owens, D. Luebke, N. Govindaraju, M. Harris, J. Krüger, A. E. Lefohn, and T. J. Purcell. A survey of general-purpose computation on graphics hardware. *Computer Graphics Forum*, 26(1):80–113, Mar. 2007. DOI: 10.1111/j.1467-8659.2007.01012.x.

[188] K. L. Palmerius, M. Cooper, and A. Ynnerman. Haptic rendering of dynamic volumetric data. *IEEE Transactions on Visualization & Computer Graphics*, 2007. To be published.

[189] J. P. Pedersen. Percutaneous nephrostomy guided by ultrasound. *Journal of Urology*, 112(2):157–159, 1974.

[190] P. Perona and J. Malik. Scale-space and edge detection using anisotropic diffusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12:629–639, 1990.

[191] J. S. Perrin, A. Lacey, R. Laitt, A. Jackson, and N. W. John. A visualization system for the clinical evaluation ofcerebral aneurysms from MRA data. In *Short Presentations Proceedings, Eurographics 2001*, Manchester, Sept. 2001.

[192] E. Persson. Framebuffer objects. ATI Radeon SDK documentation paper's collection, ATI Technologies, Inc., 2005.

[193] V. K. Peter, J. Ward, K. Sherman, R. Phillips, and D. Wills. Force torque measurements during a routine knee arthroscopy: *in vivo* study. In *EFORT 6th Congress*, Helsinki, Finland, June 2003.

[194] A. Petersik, B. Pflesser, U. Tiede, and K.-H. Höhne. Haptic rendering of volumetric anatomic models at sob-voxel. In *Proc. of the Eurohaptics Conference*, pages 182–184, 2001.

[195] D. L. Pham, C. Xu, and J. L. Prince. Current methods in medical image segmentation. *Annual Review of Biomedical Engineering*, 2:315–337, Aug. 2000. DOI: 10.1146/annurev.bioeng.2.1.315.

[196] B. T. Phong. Illumination for computer generated pictures. *Commun. ACM*, 18(6):311–317, 1975.

[197] Pov-ray. `<http://www.povray.org/>` (accessed 28th October 2007).

[198] H. Prautzsch, W. Boehm, and M. Paluszny. *Bezier and B-spline Techniques*. Springer, 2002. ISBN: 3540437614.

[199] B. Preim and D. Bartz. *Visualization in Medicine: Theory, Algorithms, and Applications*. The Morgan Kaufmann Series in Computer Graphics. Morgan Kaufmann, June 2007. ISBN: 978-0-12-370596-9.

[200] Pressure Profile Systems. Website. `<http://www.pressureprofile.com/>` (accessed 3rd October 2006).

[201] T. J. Purcell, I. Buck, W. R. Mark, and P. Hanrahan. Ray tracing on programmable graphics hardware. In *SIGGRAPH '02: Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, pages 703–712, New York, NY, USA, 2002. ACM Press. DOI: 10.1145/566570.566640.

[202] Python Software Foundation. The Python Programming Language. `<http://www.python.org/>` (accessed 15th December 2006).

[203] R3vis Corporation. RM Scene Graph, the Multi-platform OpenGL Scene Graph API. `<http://www.r3vis.com/>` (accessed 13th December 2006).

[204] M. Raspolli, C. A. Avizzano, G. Facenza, and M. Bergamasco. HERMES: an angioplasty surgery simulator. In *First Joint Eurohaptics Conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems, 2005. WHC 2005.*, pages 148–156, Los Alamitos, CA, USA, Mar. 2005. IEEE Computer Society.

[205] Reachin Technologies AB. Reachin API. `<http://www.reachin.se/products/ReachinAPI/>` (accessed 13th December 2006).

[206] Reachin Technologies AB. Website. `<http://www.reachin.se/>` (accessed 18th December 2006).

[207] W. T. Reeves, D. H. Salesin, and R. L. Cook. Rendering antialiased shadows with depth maps. *SIGGRAPH Computer Graphics*, 21(4):283–291, 1987.

[208] M. Renz, C. Preusche, M. Pötke, H.-P. Kriegel, and G. Hirzinger. Stable haptic interaction with virtual environments using an adapted voxmap-pointshell algorithm. In *Proc. of the Eurohaptics Conference*, pages 149–154, 2001.

[209] R. A. Robb, J. F. Greenleaf, E. L. Ritman, S. A. Johnson, J. D. Sjostrand, G. T. Herman, and E. H. Wood. Three-dimensional visualization of the intact thorax and contents: a technique for cross-sectional reconstruction from multiplanar x-ray views. *Computers and Biomedical Research, an International Journal*, 7(4):395–419, Aug. 1974.

[210] J. B. T. M. Roerdink and A. Meijster. The watershed transform: Definitions, algorithms and parallelization strategies. *Fundamenta Informaticae*, 41:187–228, 2001.

[211] J. Rohlf and J. Helman. IRIS performer: a high performance multiprocessing toolkit for real-time 3D graphics. In *SIGGRAPH '94: Proceedings of the 21st annual conference on Computer graphics and interactive techniques*, pages 381–394, New York, NY, USA, 1994. ACM Press. DOI: 10.1145/192161.192262.

[212] K. Rosendahl, O. G. Aasland, A. Aslaksen, A. Nordby, K. Brabrand, and V. Akre. How is the specialist training in radiology? *Tidsskrift for den Norske Laegeforening*, 120(19):2289–2292, Aug. 2000.

[213] R. J. Rost. *OpenGL Shading Language*. Addison-Wesley Professional, 2nd edition, Jan. 2006. ISBN: 0-321-33489-2.

[214] S. M. Rubin and T. Whitted. A 3-dimensional representation for fast rendering of complex scenes. In *SIGGRAPH '80: Proceedings of the 7th annual conference on Computer graphics and interactive techniques*, pages 110–116, New York, NY, USA, 1980. ACM Press. DOI: 10.1145/800250.807479.

[215] D. C. Ruspini, K. Kolarov, and O. Khatib. The haptic display of complex graphical environments. In *SIGGRAPH '97: Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, pages 345–352, New York, NY, USA, 1997. ACM Press/Addison-Wesley Publishing Co. DOI: 10.1145/258734.258878.

[216] G. Sakas, M. Grimm, and A. Savopoulos. Optimized maximum intensity projection. In *Proceedings of 5th EUROGRAPHICS Workshop on Rendering Techniques*, pages 55–63, Dublin, Ireland, 1995.

[217] J. K. Salisbury and M. A. Srinivasan. Phantom-based haptic interaction with virtual objects. *IEEE Computer Graphics and Applications*, 17(5):6–10, 1997. DOI: 10.1109/MCG.1997.10014.

[218] K. Salisbury. Haptics: The technology of touch, Nov. 1995. HPCwire Special. Nov. 10, 1995. Van Dam, Andries. Post-Wimp User Interfaces: The Human Connection.

[219] K. Salisbury, D. Brock, T. Massie, N. Swarup, and C. Zilles. Haptic rendering: programming touch interaction with virtual objects. In *SI3D '95: Proceedings of the 1995 symposium on Interactive 3D graphics*, pages 123–130, New York, NY, USA, 1995. ACM.

[220] K. Salisbury, F. Conti, and F. Barbagli. Haptic rendering: Introductory concepts. *IEEE Computer Graphics and Applications*, 24(2):24–32, Mar/Apr 2004. DOI: 10.1109/MCG.2004.10030.

[221] A. Salustri and J. R. T. C. Roelandt. Ultrasonic three-dimensional reconstruction of the heart. *Ultrasound in Medicine & Biology*, 21(3):281–293, 1995. DOI: 10.1016/0301-5629(94)00125-W .

[222] F. Salvat, J. M. Fernández-Varea, and J. Sempau. PENELOPE-2006: A code system for Monte Carlo simulation of electron and photon transport. Technical Report NEA No. 6222, Nuclear Energy Agency, OECD, 2006. Workshop Proceedings Barcelona, Spain 4-7 July 2006.

[223] Y. Sato, N. Shiraga, S. Nakajima, S. Tamura, and R. Kikinis. Local maximum intensity projection (LMIP): A new rendering method for vascular visualization. *Journal of Computer Assisted Tomography*, 22(6):912–917, 1998.

[224] H. Scharsach. Advanced GPU raycasting. In *Proceedings of CESCG 2005*, pages 69–76, 2005.

[225] M. Schijven and J. Jakimowicz. Virtual reality surgical laparoscopic simulators. *Surgical Endoscopy*, 17(12):1943–1950, Dec. 2003. DOI: 10.1007/s00464-003-9052-6.

[226] M. A. Schill, S. F. F. Gibson, H. J. Bender, and R. Manner. Biomechanical simulation of the vitreous humor in the eye using an enhanced chainmail algorithm. In *Proceedings Medical Image Computation and Computer Assisted Interventions (MICCAI)*, pages 679–687, 1998.

[227] A. G. Schreyer and S. K. Warfield. *3D image processing: techniques and clinical applications*, chapter Surface Rendering, pages 31–34. Springer, 2002. D. Caramella and C. Bartholozz, editors. ISBN: 3540674705.

[228] W. Schroeder, K. Martin, and B. Lorensen. *VTK Textbook: The Visualization Toolkit, An Object-Oriented Approach To 3D Graphics*. Kitware, Inc., 3rd edition, 2004. ISBN: 1-930934-12-2.

[229] W. J. Schroeder, J. A. Zarge, and W. E. Lorensen. Decimation of triangle meshes. In *SIGGRAPH '92: Proceedings of the 19th annual conference on Computer graphics and interactive techniques*, volume 26 of *Computer Graphics*, pages 65–70. ACM, July 1992. DOI: 10.1145/133994.134010.

[230] E. Seeram. *Computed Tomography: Physical Principles, Clinical Applications, and Quality Control*. Saunders, 3rd edition edition, 2007. ISBN: 1416028951.

[231] M. Segal, C. Korobkin, R. van Widenfelt, J. Foran, and P. Haeberli. Fast shadows and lighting effects using texture mapping. In *SIGGRAPH '92: Proceedings of the 19th annual conference on Computer graphics and interactive techniques*, pages 249–252, New York, NY, USA, 1992. ACM Press.

[232] S.-I. Seldinger. Catheter replacement of the needle in percutaneous arteriography; a new technique. *Acta radiologica*, 39(5):368–376, May 1953.

[233] SensAble Technologies, Inc. Website. <http://www.sensable.com/> (accessed 9th August 2006).

[234] SensAble Technologies, Inc. Specifications for the PHANTOM Desktop and PHANTOM Omni haptic devices. Available at <http://www.sensable.com/documents/documents/PDesktop_POmni_Specifications.pdf> 20th July 2007), Jan. 2006.

[235] SensAble Technologies, Inc. Specifications for the PHANTOM Premium 1.5/6DOF and 1.5 High Force/6DOF haptic devices. Available at <http://www.sensable.com/documents/documents/Premium_1.5_6DOF.pdf> (accessed 20th July 2007), Jan. 2006.

[236] SensAble Technologies, Inc. Stylus adapter kit for PHANTOM Desktop haptic device. Available at <http://www.sensable.com/70003600262933131l796389091/Link.htm> (accessed 1st November 2006), Jan. 2006.

[237] SenseGraphics AB. H3D. <http://www.h3d.org/> (accessed 13th December 2006).

[238] SenseGraphics AB. Website. <http://www.sensegraphics.com/> (accessed 18th December 2006).

[239] SGI. Opengl inventor. <http://oss.sgi.com/projects/inventor/> (accessed 13th December 2006).

[240] SGI. Opengl performer. <http://www.sgi.com/products/software/performer/> (accessed 13th December 2006).

[241] M. G. Silk. *Ultrasonic Transducers for Nondestructive Testing*. Adam Hilger, Dec. 1984. ISBN: 0852744366.

[242] Simbionix. Website. <http://www.simbionix.com/> (accessed 9th August 2006).

[243] C. Simone and A. Okamura. Modeling of needle insertion forces for robot-assisted percutaneous therapy. In *Proceedings of the 2002 IEEE International Conference on Robotics and Automation (ICRA '02)*, pages 2085–2091, Washington DC, May 2002.

[244] F. D. Skidmore. Junior surgeons are becoming deskilled as result of calman proposals. *British medical journal*, 327:1032–1037, Nov. 1997. DOI: 10.1136/bmj.327.7422.1032.

[245] M. Sklair-Levy, A. Polliack, D. Shaham, Y. H. Applbaum, S. Gillis, D. Ben-Yehuda, Y. Sherman, and E. Libson. CT-guided core-needle biopsy in the diagnosis of mediastinal lymphoma. *European Radiology*, 10(5):714–718, Apr. 2000. 10.1007/s003300050991.

[246] M. A. Srinivasan. What is haptics? Available at <http://www.sensable.com/3314016002829118858576169869/Link.htm> (accessed 24th June 2007).

[247] M. A. Srinivasan and C. Basdogan. Haptics in virtual environments: Taxonomy, research status, and challenges. *Computer & Graphics*, 21(4):393–404, 1997. DOI: 10.1016/S0097-8493(97)00030-7.

[248] D. P. Stables, N. J. Ginsberg, and M. L. Johnson. Percutaneous nephrostomy: a series and review of the literature. *American journal of roentgenology*, 130(1):75–82, 1978.

[249] Stanford Artificial Intelligence Laboratory (SAIL). Website. <http://ai.stanford.edu/> (accessed 18th December 2006).

[250] P. R. Stepanishen. The time-dependent force and radiation impedance on a piston in a rigid infinite planar baffle. *Journal of the Acoustical Society of America*, 49(3B):841–849, Mar. 1971.

[251] P. R. Stepanishen. Transient radiation from pistons in a infinite planar baffle. *Journal of the Acoustical Society of America*, 49(5B):1627–1638, 1971.

[252] B. Stroustrup. *The C++ Programming Language*. Addison Wesley, Reading Mass. USA, special edition, 2000. ISBN: 0-201-70073-5.

[253] K. Suffern. *Ray Tracing from the Ground Up*. A K Peters, Ltd., July 2007. ISBN: 1568812728.

[254] P. Suortti and W. Thomlinson. Medical applications of synchrotron radiation. *Physics in Medicine and Biology*, 48(13):R1–R35, July 2003. DOI: 10.1088/0031-9155/48/13/201.

[255] J. Sweeney and K. Mueller. Shear-warp deluxe: the shear-warp algorithm revisited. In *VISSYM '02: Proceedings of the symposium on Data Visualisation 2002*, pages 95–104, Aire-la-Ville, Switzerland, Switzerland, 2002. Eurographics Association.

[256] N. Taffinder, I. McManus, J. Jansen, R. Russell, and A. Darzi. An objective assessment of surgeons' psychomotor skills: validation of the MIST-VR laparoscopic simulator. *British Journal of Surgery*, 85(suppl 1)(75), 1998.

[257] N. Taffinder, C. Sutton, R. J. Fishwick, I. C. McManus, and A. Darzi. 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. Westwood, H. M. Hoffman, D. Stredney, and S. J. Weghorst, editors, *Medicine Meets Virtual Reality 1998*, volume 50 of *Studies in health technology and informatics*, pages 124–130. IOS Press, 1998.

[258] Technical University of Denmark. Field ii simulation program. Website. `<http://server.oersted.dtu.dk/personal/jaj/field/>` (accessed 29th October 2007).

[259] The Khronos Group. OpenGL Mount Evans. Siggraph 2007 - OpenGL Birds of a Feather (BOF) presentation, Aug. 2007. `<http://www.khronos.org/developers/library/2007_08_siggraph/siggraph%202007%20bof-OpenGL%20Mount%20Evans.pdf>` (accessed 18th August 2007).

[260] I. The MathWorks. Matlab. `<http://www.mathworks.com/products/matlab/>` (accessed 28th October 2007).

[261] T. Totsuka and M. Levoy. Frequency domain volume rendering. In *SIGGRAPH '93: Proceedings of the 20th annual conference on Computer graphics and interactive techniques*, pages 271–278, New York, NY, USA, 1993. ACM Press. DOI: 10.1145/166117.166152.

[262] G. E. Tupholme. Generation of acoustic pulses by baffled plane pistons. *Mathematika*, 16:209–224, 1969.

[263] F. P. Vidal. Modelling the response of x-ray detectors and removing artefacts in 3D tomography. Master's thesis, INSA-Lyon Scientific and Technical University, France, CNDRI, Laboratory of Nondestructive Testing Using Ionizing Radiations, 2003.

[264] F. P. Vidal, F. Bello, K. W. Brodlie, N. W. John, D. Gould, R. Phillips, and N. J. Avis. Principles and applications of computer graphics in medicine. *Computer Graphics Forum*, 25(1):113–137, Mar. 2006. DOI: 10.1111/j.1467-8659.2006.00822.x.

[265] F. P. Vidal, N. Chalmers, D. A. Gould, A. E. Healey, and N. W. John. Developing a needle guidance virtual environment with patient-specific data and force feedback. In H. U. Lemke, K. Inamura, K. Doi, M. W. Vannier, and A. G. Farman, editors, *CARS 2005: Computer Assisted Radiology and Surgery*, volume 1281 of *International Congress Series*, pages 418–423. Elsevier, May 2005. DOI: 10.1016/j.ics.2005.03.200.

[266] F. P. Vidal, N. W. John, and R. M. Guillemot. Interactive physically-based x-ray simulation: CPU or GPU? In J. D. Westwood, R. S. Haluck, H. M. Hoffman, G. T. M. R. Phillips, R. A. Robb, and K. G. Vosburgh, editors, *Medicine Meets Virtual Reality 15*, volume 125 of *Studies in Health Technology and Informatics*, pages 479–481. IOS Press, Feb. 2007.

[267] F. P. Vidal, N. W. John, A. E. Healey, and D. A. Gould. Simulation of ultrasound guided needle puncture using patient specific data with 3D textures and volume haptics. *Computer Animation and Virtual Worlds*, 2007. To be published. DOI: 10.1002/cav.217.

[268] P. F. Villard, G. Fournier, M. Beuve, and B. Shariat. Visualisation of physical lung simulation: an interactive application to assist physicians. In *IEEE Information Visualisation*, pages 65–70, London, UK, 2006.

[269] VRlab, Umeå University. OpenSceneGraph Haptic Library (osgHaptics. <http://www.vrlab.umu.se/research/osgHaptics/> (accessed 13th December 2006).

[270] VRlab, Umeå University. Website. <http://www.vrlab.umu.se/> (accessed 18th December 2006).

[271] W3C. Extensible Markup Language (XML). <http://www.w3.org/XML/> (accessed 13th December 2006).

[272] J. Wang, A. Sun, Y. Li, and H. Liu. Programmable gpus: New general computing resources available for desktop grids. In *Fifth International Conference on Grid and Cooperative Computing (GCC'06)*, pages 46–49, Los Alamitos, CA, USA, 2006. IEEE Computer Society.

[273] A. Watt. *3D Computer Graphics*. Addison-Wesley, 3rd edition, 2000. ISBN: 0-201-39855-9.

[274] Web3D Consorsium. Medx3d: Standards enabled desktop medical 3d. Website. <http://www.web3d.org/x3d/workgroups/medical/> (accessed 24th October 2007).

[275] Web3D Consortium. What is X3D? <http://www.web3d.org/about/overview/> (accessed 13th December 2006).

[276] J. Weickert. *Anisotropic Diffusion In Image Processing*. ECMI Series. B. G. Teubner, 1998. ISBN: 3-519-02606-6.

[277] L. Westover. Interactive volume rendering. In *VVS '89: Proceedings of the 1989 Chapel Hill workshop on Volume visualization*, pages 9–16, New York, NY, USA, 1989. ACM Press. DOI: 10.1145/329129.329138.

[278] J. J. Wild. The use of ultrasonic pulses for the measurement of biologic tissues and the detection of tissue density changes. *Surgery*, 27:183–188, Feb. 1950.

[279] E. J. N. Wilson. Fifty years of synchrotrons. In *Proceedings of the 1996 European Particle Accelerator Conference (EPAC'96)*, pages 135–139, Sitges, Spain, June 1996. CERN, Geneva, Switzerland.

[280] Xitact. Intravascular procedures. Website. <http://www.xitact.com/products/int/> (accessed 9th August 2006).

[281] T. S. Yoo, editor. *Insight into Images: Principles and Practice for Segmentation, Registration, and Image Analysis*. A. K. Peters, Ltd., 2004. ISBN: 1-56881-217-5.

[282] C. Zeller. Cloth simulation on the GPU. In *SIGGRAPH '05: ACM SIGGRAPH 2005 Sketches*, page 39, New York, NY, USA, 2005. ACM Press.

[283] Y. Zhang, M. Brady, and S. Smith. Segmentation of brain MR images through a hidden markov random field model and the expectation-maximization algorithm. *IEEE Transactions on Medical Imaging*, 20(1):45–57, Jan. 2001.

[284] Y. Zhu, D. Magee, R. Ratnalingam, and D. Kessel. A virtual ultrasound imaging system for the simulation of ultrasound-guided needle insertion procedures. In *Proc. Medical Image Understanding and Analysis (MIUA)*, volume 1, pages 61–65, July 2006.

[285] F. Zonneveld. The role of medical imaging in the management and diagnostic assessment of trauma. *Medica Mundi*, 40(3):134–150, 1995.

[286] K. J. Zuiderveld, A. H. J. Koning, and M. A. Viergever. Techniques for speeding up high-quality perspective maximum intensity projection. *Pattern Recognition Letters*, 15(5):507–517, may 1994. DOI: 10.1016/0167-8655(94)90142-2.