IMPERIAL COLLEGE LONDON

Modelling and Simulation of Flexible Instruments for Minimally Invasive Surgical Training in Virtual Reality

Przemyslaw Korzeniowski
Department of Surgery and Cancer

May 2015

A thesis submitted in partial fulfilment of requirements for the degree of
Doctor of Philosophy at Imperial College London

ABSTRACT

Improvements in quality and safety standards in surgical training, reduction in training hours and constant technological advances have challenged the traditional apprenticeship model to create a competent surgeon in a patient-safe way. As a result, pressure on training outside the operating room has increased. Interactive, computer based Virtual Reality (VR) simulators offer a safe, cost-effective, controllable and configurable training environment free from ethical and patient safety issues.

Two prototype, yet fully-functional VR simulator systems for minimally invasive procedures relying on flexible instruments were developed and validated. *NOViSE* is the first force-feedback enabled VR simulator for Natural Orifice Transluminal Endoscopic Surgery (NOTES) training supporting a flexible endoscope. *VCSim3* is a VR simulator for cardiovascular interventions using catheters and guidewires. The underlying mathematical model of flexible instruments in both simulator prototypes is based on an established theoretical framework – the Cosserat Theory of Elastic Rods. The efficient implementation of the Cosserat Rod model allows for an accurate, real-time simulation of instruments at haptic-interactive rates on an off-the-shelf computer. The behaviour of the virtual tools and its computational performance was evaluated using quantitative and qualitative measures. The instruments exhibited near sub-millimetre accuracy compared to their real counterparts. The proposed GPU implementation further accelerated their simulation performance by approximately an order of magnitude.

The realism of the simulators was assessed by face, content and, in the case of *NOViSE*, construct validity studies. The results indicate good overall face and content validity of both simulators and of virtual instruments. *NOViSE* also demonstrated early signs of construct validity. VR simulation of flexible instruments in *NOViSE* and *VCSim3* can contribute to surgical training and improve the educational experience without putting patients at risk, raising ethical issues or requiring expensive animal or cadaver facilities. Moreover, in the context of an innovative and experimental technique such as NOTES, *NOViSE* could potentially facilitate its development and contribute to its popularization by keeping practitioners up to date with this new minimally invasive technique.

Key words: **Catheter, Coronary angioplasty, Cosserat Rod, Flexible endoscope, Guidewire, Medical simulator, Natural Orifice Transluminal Endoscopic Surgery, Virtual Reality**

ACKNOWLEDGEMENTS

First and foremost, I would like to express my gratitude to my supervisor, **Dr Fernando Bello** for giving me the opportunity to be a part of his research team and for leading me throughout my studies and work. Fernando, your commitment and given support were vital at every stage of my work. I am grateful for your invaluable feedback, constructive comments, practical suggestions and critical thinking without which I would surely have been lost. Without your dedication and wisdom, it would be extremely hard, if not impossible, for me to achieve all the targets set. I am very grateful for your precious time you spent on reading and correcting my articles and reports including this thesis as well as for securing funding for my projects throughout all these years.

I would like to extend my gratitude to clinical experts who helped me with my projects, especially to **Mr Mikael H Sodergren**, **Dr Sujatha Kesavan** and **Dr Emmanuel Ako**. Thank you for your expertise, insight and time.

I feel indebted to all of the clinical participants who took the time from their busy schedules to participate in my research. Without their involvement and feedback, all the essential validation studies and, as a result, this thesis would not have been possible.

I would like to express special thanks to **Dr Vincent Luboz,** for introducing me to the challenging yet fascinating field of medical simulation, and **Prof Roger Kneebone** for giving me the opportunity to collaborate with his group in public engagement and science communication events. This surely helped to improve my soft and acting skills. I would also like to thank **Prof Aleksander Nawrat** for pointing me long time ago in, as it now turned out, the right direction.

I am also thankful to the administration staff at the Department of Surgery and Cancer at St. Mary's: **Mr Duncan Boak**, **Mr Taliesin Williams**, **Mr Stephen Marchington**, **Ms Megan Roy** and **Mrs Louise Dawson** for all your assistance provided.

I would also like to acknowledge the generous support of the **Engineering and Physical Sciences Research Council** who largely funded my projects.

I could not finish without acknowledging my great past and present colleagues from the neighbouring desks: **Alejandro, Alastair, Diane, Francisco, Hafiz, Niels** and **Pete** as well as from the office next-door: **Ana-Rita**, **Alex, Jason, Jessica, Jimmy, Laura, Matt, Sharon, Stella, Tanika** and **Zinah.** I feel very lucky to have met and worked with all of you.

Many thanks go to my friends and colleagues in Poland and in the UK with whom I spent this challenging, but at the same time, very exciting period of my life.

I am extremely grateful to my family: my parents **Bożena** and **Henryk**, and my brother **Klaudiusz** for believing in me and their encouragements to pursue my interests. Without your unconditional support it would be impossible for me to undertake, first the MSc and later, the PhD studies in the UK.

Finally, a special and warmest thank you to **Agnieszka,** my fiancée, for all her love and support she has given me. I am very lucky to have you in my life. I dedicate this thesis to you because, by the time of my viva, you will be my wife. I look forward to our lifelong journey.

DECLARATION OF ORIGINALITY

This document consists of research work conducted in the Department of Surgery and Cancer at Imperial College London between 2011 and 2015. I declare that the work presented is my own and this thesis contains no material previously published or written by another person, except where specifically acknowledged in the text.

Przemyslaw Korzeniowski

COPYRIGHT DECLARATION

Table of Contents

LIST OF FIGURES

LIST OF TABLES

Chapter 1

INTRODUCTION

This chapter gives an overview of the research addressed in this thesis. Section 1.1 defines the application domain by introducing the concepts of Minimally Invasive Surgery, surgical training methods and computer-based simulation systems. The motivation behind the work and research is given in Section 1.2. The aims and objectives of the research are summarized in Section 1.3, followed by the specific contributions to the state of the art listed in Section 1.4. The related publications are listed in Section 1.5 and the public engagement events in Section 1.6. The outline of the thesis is presented in Section 1.7.

1.1 APPLICATION DOMAIN AND HYPOTHESIS

The application domain are computer-based surgical training systems for minimally invasive procedures relying on flexible surgical instruments. The hypothesis is that the research and work presented in this thesis contribute towards an effective training of basic manual skills for flexible endoscopy and endovascular interventions without compromising patient safety.

1.1.1 Minimally Invasive Surgery

Traditionally, in order to carry out a surgical procedure, a surgeon had to perform an incision in the patient's body (open surgery). The incision had to be large enough to permit the unrestricted visibility and direct access to the organs with surgical tools. For example, a laparotomy (Figure 1.1 left) is an incision through the abdomen to gain access into the abdominal cavity. It normally preceded standard surgical procedures such as cholecystectomy (gallbladder removal) or appendectomy (appendix removal). A thoracotomy is an incision into the pleural space of the chest to gain access, usually, to the heart or lungs. Post-operative incisions can be difficult to deal with. They cause pain, take time to heal and can result in complications such as haemorrhage, infections or adhesions (internal scar tissue that connects tissues not normally connected).

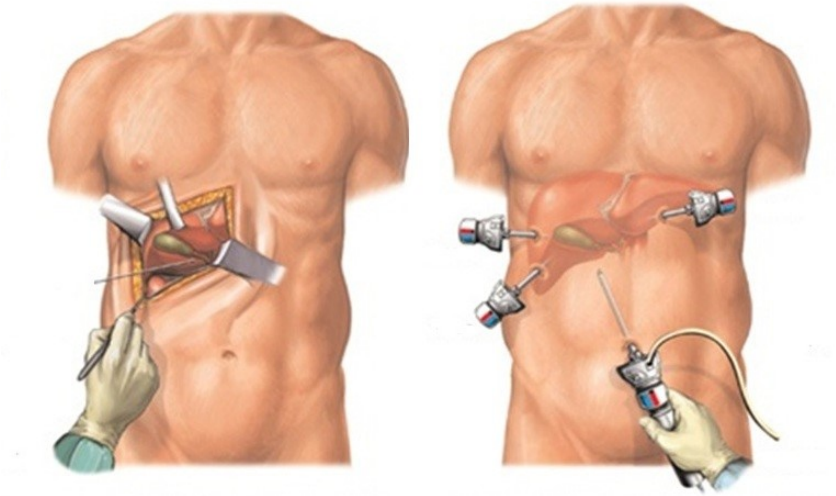


Figure 1.1: Examples of open (left) and minimally invasive surgery (right) taken from (www.gallbladder-surgery.org).

In order to reduce post-operative pain and infection risks, minimally invasive surgical (MIS) techniques have been developed (Wickham, 1987). MIS replaces relatively large incisions by much smaller ones serving as entry points for specialized surgical instruments, rigid or flexible. The instruments range from thin plastic tubes – catheters – navigated in the patient's vascular system using external visualization such as computer tomography (CT), to complex devices - endoscopes - equipped with a light source, a lens system or a LCD camera allowing for visualization of the patient's internal anatomy. Regardless of the picture acquisition method, the visual feed is commonly presented to a practitioner on a separate screen (Figure 1.2). Therefore, surgeons manipulate the instruments via an indirect view of the surgical site.

Figure 1.2: Minimally invasive procedures: laparoscopy (left) and cardiovascular intervention (right).

In fact, over the last 25 years, MIS has become the de facto standard for some operations, most notably, for abdominal and cardiovascular procedures. Smaller incisions result in decreased recovery time, reduced postoperative pain, lower morbidity and mortality (Wickham, 1987). This leads to a shorter hospitalization time and, in a broader perspective, translates into lower healthcare costs and employee absenteeism (Epstein et al., 2013). MIS also improves the overall cosmetic result. The main disadvantage of MIS is for the surgeons who lose direct visualisation, as well as manual contact with the surgical site due to the use of longer instruments. As a result, gaining proficiency in MIS requires an intensive surgical training.

In order to push MIS techniques further towards truly non-invasive surgery, surgeons have recently started experimenting with a novel technique whereby a flexible endoscope is inserted via a natural orifice to gain access to the abdominal cavity, leaving no external scars. This emerging technique is known as Natural Orifice Transluminal Endoscopic Surgery (NOTES). Since it eliminates external postoperative wounds, it is argued that NOTES may further reduce operation trauma, recovery time, clinical costs and improve overall cosmetic results.

1.1.2 Surgical training

In the beginning of the previous century, William Halsted outlined an apprenticeship model of surgical training widely adapted to this day (Halsted, 1904). Halsted's mentor-learner model was created in the times when society was tolerant of medical errors, including the ones committed by surgical trainees. The model is based on a high-volume, hands-on training with gradually decaying level of supervision, until the trainee is judged by his mentor competent enough to operate on his/her own. Improvements in quality and safety standards in surgical training yielded the apprenticeship model insufficient to create a competent surgeon in a patient-safe way. As a result, pressure on training outside the operating room (OR) has increased. A recent systematic review (Zendejas et al., 2013) shifted the conversation regarding the surgical skills training outside the operating room from *"Is it effective?"* to *"How can it be most effective?"* (Selzer and Dunnington, 2013). Based on simulation-based training for laparoscopic surgery, Zendejas et al. concluded that "*simulation-based laparoscopic surgery training of health professionals has large benefits when compared with no intervention and is moderately more effective than nonsimulation instruction*" (Zendejas et al., 2013). However, gaining core surgical skills on animals or cadavers is expensive and brings ethical issues, thus restricting their use in everyday training. Using inexpensive, low-fidelity task physical trainers (Figure 1.3) can provide effective training of the key elements of the procedure, but lacks the real-life effect of a complete surgery.

Figure 1.3: Two examples of low-fidelity box trainers: Simulab's TraumaMan (www.simulab.com) on the left and cheap laparoscopic box trainer designed by Ali Bahsoun at King's College London (www.kingshealthpartners.org) on the right.

Moreover, animals and cadavers, as well as foam, silicon or plastic parts used in task trainers, have a lack of physiological behaviour and different bio-mechanical properties compared to human tissue. Hence, these methods do not provide sufficient realism. Finally, to be most efficient, they require feedback from an expert at different training stages.

1.1.3 Computer-based training systems

An alternative approach is an interactive computer-based simulation enabling safe training on a virtual patient (Gallagher et al., 2005). Such systems, often referred to as virtual reality (VR) simulators (Figure 1.4), typically consist of a 2D or 3D display, a computer running the simulation software, and a physical human-computer interface device mimicking the surgical instruments. The device tracks the manipulation of the instruments and often can recreate the sense of touch by providing force-feedback to the user (a haptic device). The software is responsible for taking input from the input device, simulating the interactions between the instruments and the virtual anatomy, rendering the 3D image of the surgical site and, if supported, calculating the forces sent to the user via the haptic device. Additionally, the software can record, analyse and store user performance.

High-fidelity VR simulators have a number of advantages over the aforementioned methods of surgical training. They provide a safe, controllable and configurable

training environment free from ethical issues in which clinicians can repetitively practice without putting patients at risk. VR simulators have been expected to become an important part of surgical training since the early 1990s (Satava, 1993). Whilst their initial cost might seem expensive at first, VR simulators can in fact be cost-effective when considering the wider economic benefits of better-trained surgeons and resource optimisation (Bridges and Diamond, 1999). Studies show savings on instructor time, error reduction and faster completion times (Seymour et al., 2002, Aggarwal et al., 2007). Moreover VR simulators can be applied to explore new ways of performing a surgery, prototyping medical equipment or to become familiar with new surgical techniques or new surgical devices (Punak and Kurenov, 2011b). This is particularly interesting in the context of new experimental methods such as the aforementioned NOTES. Recent reviews show that, although VR simulation is now successfully used in various surgical specialities, there is still enormous potential for further development (Gallagher and Traynor, 2008, de Visser et al., 2011, de Montbrun and Macrae, 2012).

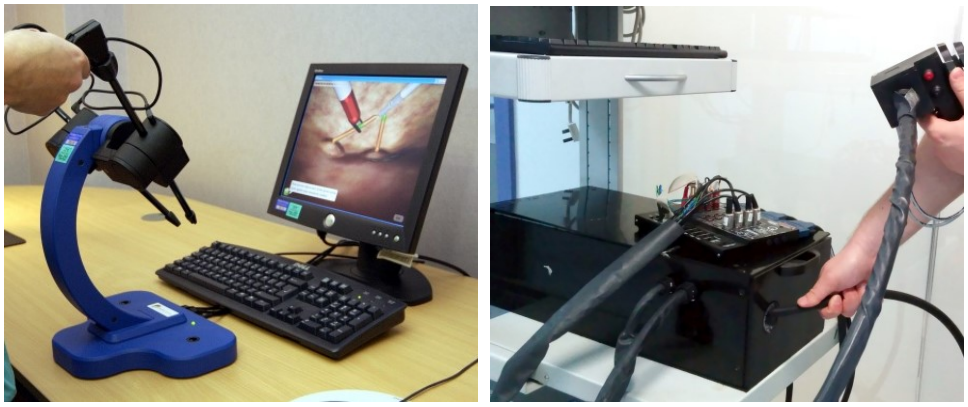


Figure 1.4: Two examples of VR simulators: On the left: LapSim simulator (www.lapsim.com) using Immersion's Virtual Laparoscopic Interface device (www.imersion.com). On the right: A prototype haptic device for endoscopic procedures.

1.2 MOTIVATION

Improvements in quality and safety standards in surgical training, reduction in training hours and constant technological advances have challenged the traditional apprenticeship model to create a competent surgeon in a patient-safe way. Therefore, the pressure on training outside the operating room has increased. Interactive, computer based virtual reality (VR) simulators offer a safe, cost-effective, controllable and configurable training environment free from ethical and patient safety issues.

The design and development of a VR surgical simulator is a complex process requiring interdisciplinary knowledge from computing, engineering, physics, surgery and medicine. Despite the increasing computational power of modern computers and intensive research in the field, the implementation of simulation software responsible for the realistic recreation of surgical procedures is a challenging and open problem.

In this thesis, the emphasis is placed on simulation of flexible surgical instruments in two chosen surgical procedures: flexible endoscopes in natural orifice transluminal endoscopic surgery (NOTES), and catheters and guidewires in cardiovascular interventions. Flexible surgical instruments are difficult to model, implement and simulate in VR as they not only bend and do not stretch, but may also twist resulting in a complex looping phenomenon.

The combination of tools and actions in a NOTES procedure creates many unique challenges for simulation and, to the best of the author's knowledge, there are currently no VR simulators supporting flexible endoscope, either commercial or experimental, aimed specifically at NOTES.

During a cardiovascular procedure, catheters and guidewires can be manipulated from the patient's femoral artery via the aorta, all the way up into the coronary arteries of the heart. Real-time simulation of such long and thin instruments results in a trade-off between interactivity and accuracy. Therefore, developing

computationally efficient methods is crucial for delivering an effective training experience.

Considering the above, providing realistic VR simulation of the aforementioned procedures can contribute to surgical training and improve the educational experience without putting patients at risk, raising ethical issues or requiring expensive animal or cadaver facilities. Moreover, in the context of an innovative and experimental technique such as NOTES, it could potentially facilitate NOTES development and contribute to its popularization by keeping practitioners up to date with this novel minimally invasive method.

1.3 AIMS AND OBJECTIVES

The aim of this research project is to develop and evaluate virtual reality simulation systems for minimally invasive surgical training relying on flexible surgical tools. This involves designing and implementing software solutions combining realistic simulation, visualization and interactions. The software will be integrated with third-party haptic devices in order to deliver a complete VR training experience. Finally, the clinical realism will be assessed using quantitative and qualitative measures.

As mentioned, particular emphasis will be placed on modelling and simulation of instruments used during cardiovascular interventions and flexible endoscopy procedures such as NOTES. This is because tools used in these procedures have very different mechanical properties. The catheter and guidewire are long and thin instruments requiring subtle force and precision. On the contrary, the flexible endoscope is shorter, much thicker and heavier. It requires substantially different force and manipulation techniques. Having proven methods for simulation of tools in these two different procedures would lay the foundations for simulation of other flexible instruments and minimally invasive procedures.

The specific objectives to achieve the research aim are:

- To review software and hardware components necessary to develop modern computer based virtual reality medical simulators:
    - OUTCOME -> a review of existing solutions.

- To review existing models of one-dimensional flexible bodies (elastic rods) in order to choose the most appropriate for the simulation of virtual flexible tools in terms of both realism and performance:
    - OUTCOME -> a review of existing models.

- To adapt the chosen model(s) to fit the requirements of the software framework under development by:
    - Understanding the relations between the elastic rod model parameters and its behaviour.

  - Tuning the model implementation and parameters to achieve the required level of realism and responsiveness to user interactions.
  - Developing and customizing collision detection and response algorithms to recreate interactions between the simulated tool and the virtual environment.
    - OUTCOME -> A qualitative and quantitative description of the behaviour of the chosen model.

- To use the chosen model to develop a virtual reality simulator for flexible endoscopy and, specifically, for NOTES by:
  - Adapting the rod model to the simulation of virtual flexible endoscope consisting of flexible shaft with steerable tip, range of end effectors and camera.
  - Identifying the aspects differentiating simulation of NOTES procedures from simulation of other MIS and endoscopic procedures in terms of specific manipulative skills.
  - Adjusting the behaviour of the flexible endoscope to meet specific requirements of NOTES surgery, such as using an instrument inserted through a single hole, lack of the gastrointestinal lumen to support the endoscope, manipulations in open space of abdominal cavity, use of organs and gravity for navigation and viscerotomy sites as fulcrum points.
  - Implementing a complete NOTES procedure consisting of a set of specific tasks allowing for carrying out validation studies
  - Assessing and verifying the clinical realism of the virtual flexible endoscope using a combination of qualitative and quantitative methods.
    - OUTCOME -> A simulator of the flexible endoscope with a set of NOTES specific training tasks which clinicians agree is realistic enough to be useful for surgical training.

- To apply the chosen rod model to develop a virtual reality simulator for endovascular interventions by:
  - Adapting the model to the simulation of catheters and guidewires
  - Cooperating with clinicians in order to find a consensus of how the virtual instruments should behave during the endovascular procedure.
  - Fine-tuning the parameters of virtual catheters and guidewires in order to match their behaviour to their real-life counterparts (i.e. specific commercially available products).

- Implementing a complete endovascular procedure permitting carrying out validation studies.
- Assessing and verifying the clinical realism of virtual catheters and guidewires using a combination of qualitative and quantitative methods.
    - OUTCOME -> A simulator for cardiovascular interventions with a set of virtual catheters and guidewires, which clinicians agree is sufficiently realistic to be useful for surgical training.

Achieving the above aims and objectives will require a significant amount of iterative software development in consultation with collaborating clinicians, as well as interfacing with bespoke haptic devices. There will also be considerable opportunities to innovate and contribute to the wider field of surgical simulation through the addressing of the key research challenges mentioned above.

1.4 CONTRIBUTIONS

This thesis and research carried out makes several contributions to the field:

- Modification of a proposed Cosserat rod implementation - the CoRdE model - to improve its performance, make it stable and inextensible, hence allowing for accurate real-time simulation at haptic-interactive rates, fast response to user interactions and easy parameterization.
- Application of the model to the simulation of catheters, guidewires and flexible endoscopes.
- Development of *NOViSE* – the first VR simulator for Natural Orifice Transluminal Endoscopic Surgery. The procedure simulated is a hybrid, transgastric cholecystectomy. The simulator realism was assessed by establishing its face, content and construct validity.
- Cosserat Rod parameter optimization using genetic algorithms to match the mechanical behaviour of real catheters and guidewires.
- Development, face and content validation of *VCSim3* – a VR simulator for cardiovascular interventions.
- Efficient massively-parallel CUDA implementation of the Cosserat Rod model using inter-block synchronization.

1.5 PUBLICATIONS

Part of the research and work presented in this thesis has been published or is currently under review:

- *"Simulation of Catheters and Guidewires for Cardiovascular Interventions Using an Inextensible Cosserat Rod."*, Korzeniowski, P.; Martinez-Martinez, F.; Hald, N.; Bello, F. – Proceedings of 6th International Symposium on Biomedical Simulation (ISBMS), Lecture Notes in Computer Science, Volume 8789, Publisher: SPRINGER INT PUBLISHING AG, Pages: 112-121, ISSN: 0302-9743.

- *"NOViSE – a Virtual Natural Orifice Transluminal Endoscopic Surgery Simulator"*, Korzeniowski, P.; Barrow, A; Sodergren M. H.; Hald, N.; Bello, F. – submitted to International Journal of Computer Assisted Radiology and Surgery (IJCARS).

- *"Validation of NOViSE – a novel Natural Orifice Virtual Surgery Simulator",* Korzeniowski, P.; Brown D. C.; Sodergren M. H.; Bello, F. – submitted to Surgical Innovation.

1.6 PUBLIC ENGAGEMENT AND SCIENCE COMMUNICATION

The results of this thesis have been regularly and extensively presented to the general public, clinicians and scientists at a number of public engagement and science communication events across the UK and internationally, as well as to stakeholders and visitors to the Department of Surgery and Cancer:

- ✓ The Royal Institution of Great Britain - Lates 2015
- ✓ The Science Picnic 2014 (Warsaw, Poland, 100.000+ visitors)
- ✓ The Times Cheltenham Science Festival 2014 (45.000 visitors)
- ✓ The Big Bang Fair 2013 (ExCeL, London, 65.000 visitors)
- ✓ The Big Bang Fair 2012 (NEC, Birmingham, 49.000 visitors)
- ✓ "Teaching your eyes to feel" - The Royal Institution of Great Britain 2014
- ✓ "Teaching your fingers to see" - The Royal Institution of Great Britain 2013
- ✓ Science Museum Lates
- ✓ Natural History Museum
- ✓ Imperial Festival
- ✓ Imperial West Launch Event

Please refer to Appendix B for photographic evidence of the above events.

1.7 THESIS OUTLINE

This thesis describes the work and research done in modelling and simulation of flexible surgical instruments for minimally invasive surgery training in virtual reality.

Chapter 2 starts by presenting an overview of the history and application of simulation in training. It then focuses on virtual reality simulators and its components, such as visualization, haptic devices and approaches to tissue and instrument simulation in more detail.

Chapter 3 provides the theory behind the Cosserat Rod and one of its computer implementations – the CoRdE model. Modifications to this model which make it more suitable for application in real-time surgical simulators are proposed.

Chapter 4 introduces the first VR simulator for Natural Orifice Transluminal Endoscopic Surgery (NOTES). The chapter starts with the background on this novel technique and procedure, describing the methods applied to simulate a flexible endoscope. It then discussed the steps of the virtual procedures. Finally, results of face, content and construct validity of the simulator are presented.

In Chapter 5, the VR simulator for cardiovascular interventions is introduced. Similarly to the previous chapter, it starts with the background on the procedure itself, followed by a review of existing solutions. Methods used to simulate catheters and guidewires are then presented. At the end of the chapter, the results of face and content validation, as well as computational performance of the simulator, are given.

Chapter 6 describes a massively-parallel implementation of the Cosserat Rod using CUDA and presents a detailed performance results and analysis.

Chapter 7 summarizes the work presented in the previous chapters, discusses the research carried out, identifies limitations of the approaches used, and outlines possible future work.

Chapter 2

BACKGROUND

The aim of this chapter is to familiarize the reader with the concept of virtual reality simulation in medical and surgical training. Section 2.1 briefly discusses the use of simulation in training, its history and the technology behind it. Section 2.2 focuses on the application of simulation to medical training and education, whilst Section 2.3 concentrates on the emergence of early computer-based virtual reality simulators. Lastly, Section 2.4 reviews both hardware and software components of such simulators placing emphasis on the aspects of physically based simulation.

2.1 SIMULATION IN TRAINING

Simulation is the imitation of a real-world process or system over time (Banks, 2010). Interactive simulation is a type of simulation that requires human interaction. It can be used for training purposes, for example, to exercise motor control, decision making or communication skills. Training using interactive simulation may be preferable when it is too dangerous or too expensive to train in the real world. The use of simulation for training is currently well-established in industries such as aviation, military or astronautics (Issenberg et al., 1999).

The first known flight simulator (Figure 2.1 left) was developed over 100 years ago (1909) to help pilots operate the Antoinette plane. It consisted of a seat mounted in a pivotable half-barrel and two wheels. Assistants outside pitched and rolled the whole device according to the pilot manipulation of the wheels. In 1929, Ed Link built the *Link Trainer* (Figure 2.1 right), the first in a series of flight simulators to teach new pilots how to fly by instruments. In 1934, the Link Trainer was adapted by the US Army Air Corps as a result of a series of devastating plane crashes happening when flying in poor weather conditions. Subsequently, the Link Trainer simulator was widely used as a key pilot training aid during World War II. More than half a million pilots were trained only in the US using this simulator.

Figure 2.1: Left: The Antoinette training rig considered as the first flight simulator (1909). Right: The Link Trainer, the first widely adapted flight simulator (1930s-1950s).

Twenty five years later, in 1955, the Federal Aviation Administration (FAA) incorporated flight simulator training as a mandatory requirement for flight certification. Since then, the use of first analogue, and later digital computers, significantly increased the realism of flight simulation. NASA's Apollo simulators

were the first fully digital designs. Currently, all military and commercial pilots in the US must be trained on flight simulators. Although they are initially expensive, they have proven to be cost-effective in the long run (Rosen, 2008).

Technological progress allowed *immersive* simulation to be tested for various different purposes in other fields. One of the earliest classic examples is the *Sensorama (Hamit, 1993)* showed in Figure 2.2. It was a mechanical arcade machine dubbed "The Experience Theatre". Built in 1962 by cinematographer Morton Helig, it was intended to deliver an immersive, multi-sensory (multimodal) experience, featuring stereoscopic image and sound, tilting seat, wind and olfactory effects. The device was non-interactive, and played a pre-programmed experience, such as riding a motorcycle through the streets of Brooklyn. Viewers had to put their eyes to immovable lenses. To address this inconvenience, the concept of head-mounted displays (HMD) appeared around the same time.

Figure 2.2: Sensorama, "The Experience Theater" from 1962 (www.mortonheilig.com)

The first HMD was built by Philco Corporation in 1961 (Kalawsky, 1993). Their device displayed actual video footage from a camera mounted in another room. It used magnetic tracking to track head orientation and rotate the remote camera accordingly, thus creating a sense of *telepresence*. However, the most iconic device, which is widely considered as the first stereoscopic HMD, was the Ultimate

Display (Sutherland, 1968). It was built in 1968 by Ivan Sutherland, a pioneer in computer graphics. The device displayed a simple, computer generated 3D wireframe into the binocular display. The perspective would depend on the head orientation of the user, which was mechanically tracked and resulted in the device having considerable weight. It was so heavy that it had to be mounted on the ceiling of Sutherland's lab (Figure 2.3), hence, its nickname - the *"Sword Of Damocles"*. Sutherland was particularly interested in using his HMD to create virtual worlds (VW) that violated the rules of physical reality.

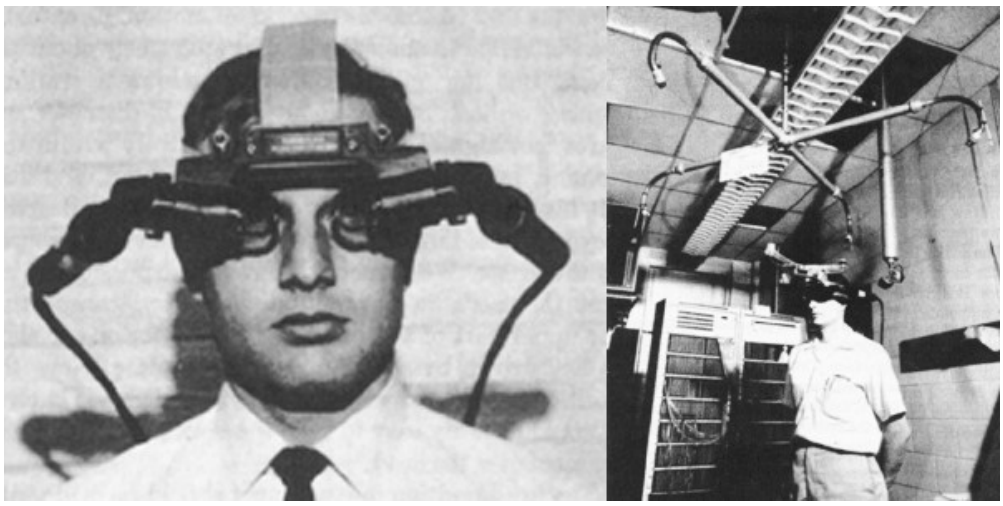


Figure 2.3: Ivan Sutherland wearing his head-mounted display (Sutherland, 1968).

Twenty years later, in 1987, Jaron Lanier coined the term "Virtual Reality" and announced the era of interactive, image-based computer simulations (Satava, 2008). His company, VPL Research, was one of the first to commercially develop VR applications and sell VR products (VRS, 2015). Most notable are *The EyePhone* – the first commercially available HMD, and *The DataGlove* – a device which uses a glove as a form of user input. Later these products were used as key components of the first VR simulator for general surgery (Satava, 1993).

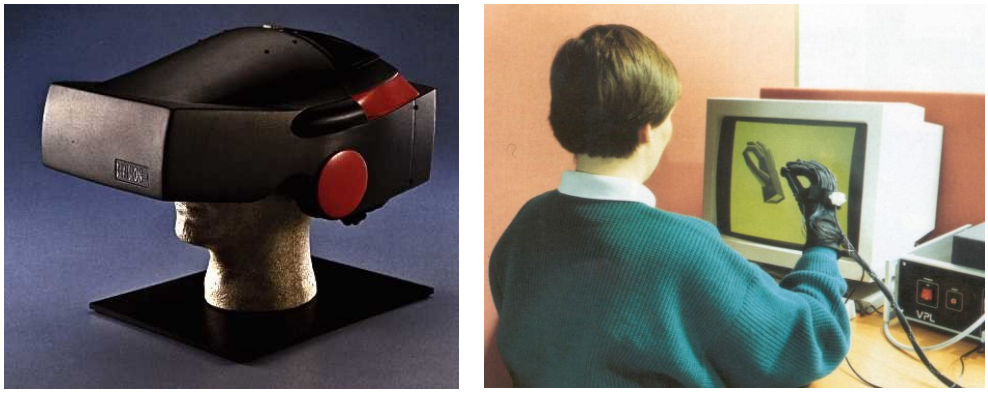


Figure 2.4: The two commercial VR products by VPL Research: The EyePhone (left) and The DataGlove (right).

Throughout the 90s, there were high expectations about VR entering the mass-market with HMDs as the focal point. However, consumer VR turned out to be ahead of its time. Both hardware and software simply could not deliver the given promises and the idea rapidly faded from the public eye for more than a decade. Given the recent progress in computer graphics, display and sensor technology, the concept of consumer VR has been brought to life once again. Currently, there are no affordable HMDs available for purchase, but a number of devices was announced for release in 2016, most notably, Oculus Rift ($599, *www.oculus.com*), HTC ViVe (*www.htcvive.com*) and *Sony Morpheus* (*www.playstation.com*). An interesting concept is to use a smartphone inserted in a low-cost head-mounted casing (e.g. *www.samsung.com/gear_vr*). Modern smartphones offer high resolution displays (e.g. QHD 1440 x 2560), necessary head-tracking capabilities and computational power sufficient to deliver a modest VR and AR experience.

Figure 2.5: Two commercial HMDs. Oculus Rift (left) and Samsung GearVR (right).

2.2 SIMULATION IN MEDICAL EDUCATION

Medicine has been using simulation for centuries in many forms such as animal models or cadavers (Rosen, 2008). More recently, in the 1960s, actors called "standardized patients" were used for the first time to train in communication and physical examination skills (Rosen, 2008, Singh et al., 2013). However, it took more than 40 years (2004) until they were incorporated into the national licensing process in the US (Singh et al., 2013). Also, in the 60s, mannequins and task trainers were introduced to medical education. *Resusci Annie* (Singh et al., 2013) was a mannequin designed to teach mouth-to-mouth and later cardiopulmonary resuscitation (CPR). An upgraded version is still in production and use. *Sim-One* (Figure 2.6) was the first computer-controlled, interactive simulated patient. Initially developed to train anaesthetists, it had a heartbeat, palpable pulse, chest respiratory movement, pupils changing size and could respond to numerous medications.

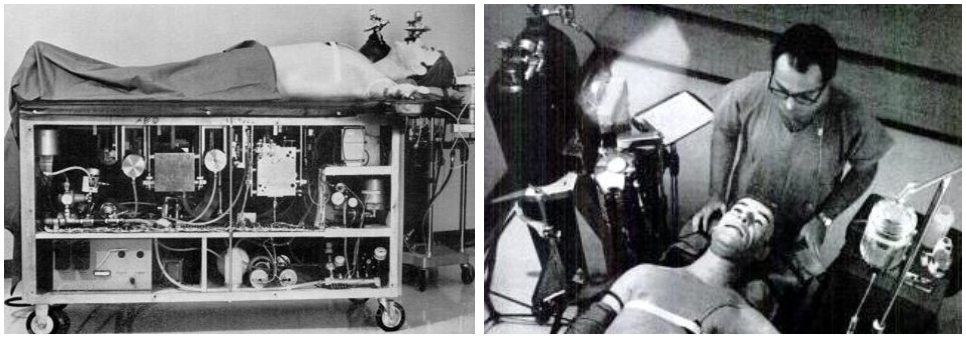


Figure 2.6: Sim One was the world's first interactive simulated patient (Life Magazine, 8 Dec 1967)

Similarly to flight simulators, the dawn of digital age allowed for further improvements in realism and enabled for simulation of different scenarios and physiologic responses. This converged with the popularization of Minimally Invasive Surgery (MIS). Further constant increase in computer power and emergence of haptic technology (Salisbury et al., 2004) resulted in the concept of medical virtual reality simulators.

2.3 VIRTUAL REALITY IN MEDICAL EDUCATION

Virtual reality (VR) simulators have been expected to become as important for surgery as flight simulators are for aviation since the 1990s (Satava, 1993). In 2001, Satava estimated that the field was in the same era as the original flight simulators developed by Ed Link and stated that "*The greatest power of virtual reality is the ability to try and fail without consequence to animal or patient. It is only through failure – and learning the cause of failure – that the true pathway to success lies*" (Satava, 2001).

One of the first medical VR simulators was developed in 1987 at Stanford University to practice Achilles' tendon repair (Delp et al., 1990) (Figure 2.7 left). The simulator could also be used for preoperative planning. It allowed to "walk" the leg and visualize the effect of the procedure on the gait. A few years later, Lanier and Satava (Satava, 1993) developed a first simulator for simplified intra-abdominal surgery.

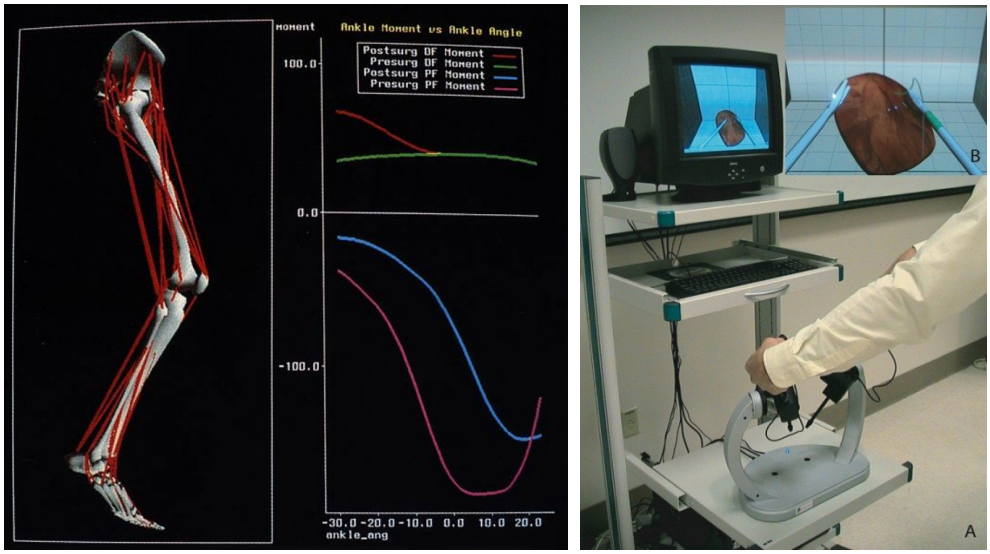


Figure 2.7: Left: the first VR simulator for Achilles' tendon repair. Right: MIST-VR, the first successful VR simulator.

The first commercially successful VR surgical simulator was the Minimally Invasive Surgery Trainer-VR or MIST-VR, (Wilson et al., 1997) by Mentice, Sweden (www.mentice.com, Figure 2.7 right). It was based on abstract graphics and consisted of fundamental laparoscopic tasks emphasising motor skills acquisition.

(Seymour et al., 2002) demonstrated its validity and estimated a 29% reduction in operating time and a 85% decrease in number of errors during gallbladder dissection in a laparoscopic cholecystectomy procedure.

Currently, there are simulators for many subspecialties, such as laparoscopic surgery (e.g LapMentor, Figure 2.8 left, www.simbionix.com), endovascular surgery (e.g. VIST, Figure 2.8 right, www.mentice.com), endoscopy (e.g. EndoSim), etc (Dunkin et al., 2007).

Figure 2.8: The state-of-the-art high-fidelity VR simulators. On the left: Lap Mentor III, laparoscopic simulator by Simbionix (www.simbionix.com). On the right: Vist-Lab, cath-lab simulation by Mentice (www.mentice.com).

Virtual reality simulators have a number of advantages over the aforementioned methods of surgical training. They offer a safe, controllable and configurable training environment free from ethical issues. VR simulators improve patient safety. Not only because patients are not at risk during actual training, but also because surgeons trained on VR simulators show higher competencies (Seymour et al., 2002, Youngblood et al., 2005).

VR simulators improve the educational experience by providing a wide selection of training scenarios diversified in terms of virtual patient's anatomy and pathologies. This overcomes the problem of waiting for a suitable real-life case, and allows for

controlled clinical exposure where trainees start with basic cases moving gradually to more complex ones when they feel confident to do so.

Training on VR simulators does not require the presence of a supervising expert. By analysing user performance in real-time, simulators can give immediate feedback during the procedure, which is crucial for efficient training (Zendejas et al., 2013). The formative and summative assessment at the end of each training session helps to track user's learning progress that may be used in the future for credentialing and certification (de Visser et al., 2011).

VR simulators have low maintenance costs and, except for calibration, practically require no preparation before or during the training session. Students can train on their own, whenever the equipment is available. The cases can be repeated countless times without generating additional costs. In contrast to mannequins and box trainers, VR simulators do not wear and tear.

Experts can also benefit from simulation by practising rare-cases to maintain and improve their skills, or even to "warm-up" before performing real surgery (Kahol et al., 2009). VR simulators can be used to explore new ways of performing a procedure or to become familiar with new surgical techniques or new surgical devices (Punak and Kurenov, 2011b). This is particularly interesting in the context of experimental techniques such as the aforementioned NOTES.

Some VR simulators can assist during pre-operative planning or intra-operative navigation (Kockro et al., 2000). By reading patient specific data obtained from medical imaging (CT or MRI), VR simulators can help to plan a surgery in order to avoid potential complications or predict the outcome.

High development costs and corresponding final high price is usually mentioned as a key disadvantage of VR simulators. However, when considering the wider economic benefits of better-trained surgeons, error reduction, faster completion times and savings on instructor time, VR simulators can in fact be cost-effective (Bridges and Diamond, 1999, Seymour et al., 2002, Aggarwal et al., 2007).

Despite the ongoing improvements, VR simulators still lack the ultimate realism as there is no substitute for the human body. Due to their complex bio-mechanical behaviour, the behaviour of tissues and organs is difficult to simulate, especially in real-time. There are also concerns about lack of face-to-face patient contact, which may result in physicians not properly developing necessary *soft skills*.

Recent reviews show that, although VR simulation is now successfully used in various surgical specialities, there is still enormous potential for further development (Gallagher and Traynor, 2008, de Visser et al., 2011, de Montbrun and Macrae, 2012)(Satava, 2013, Gallagher and Satava, 2015).

Further details on the use and advances of simulation in medical education, as well as its history can be found in (Rosen, 2008), (Satava, 2008), (de Montbrun and Macrae, 2012) and (Singh et al., 2013).

2.4 COMPONENTS OF VIRTUAL REALITY SURGICAL SIMULATORS

A VR surgical simulator typically consists of a haptic interface – a device which recreates the sense of touch by applying forces, vibrations, or motions to the user, a display, and a computer running the simulation software. Their aim is to deliver an interactive (real-time) visual and tactile experience with realism sufficient to achieve the anticipated learning outcome (Ruthenbeck and Reynolds, 2013). The software consists of a number of sub-systems, which work together to visually and physically simulate the virtual anatomy, surgical instruments, as well as interactions between them. Below, an overview of both hardware and software components of a simulator is presented.

2.4.1 3D real-time graphics

3D computer graphics (CG), animation and visualization are well-established and documented fields. In 1996, the chairman of Silicon Graphics (SGI) said that the gaming industry surpassed the military in development of high-resolution graphics. It is estimated that the video game (Figure 2.9) market revenue can exceed $100 billion worldwide by 2017 (CFN, 2014).

Figure 2.9: The screenshots from GTA3 (2001, left) and GTA4 (2008, right) video games showing the improvements in consumer computer graphics quality.

This rapid growth has resulted in fast and affordable graphics processing units (GPUs) becoming widely available. The GPUs are accessed via a low-level application programming interfaces (APIs) such as OpenGL and DirectX. On top of these APIs, a number of higher level graphics frameworks have been developed such as Open Graphics Rendering Engine (OGRE 3D, www.ogre3d.org), Unity3D (www.unity3d.com) and Unreal Engine (www.unrealengine.com), to name but a

few. They are often referred to as "*3D engines*" and allow for rapid preparation of interactive scenes consisting of 3D meshes, lighting, special effects (e.g. fog, particle effects) and cameras. The scenes can be set-up in a built-in editor, via scripting, programming in native code or any combination of these. Additionally, they offer audio, input, animation and physics support. The latter is usually done by integrating with third-party libraries (*"middleware"*) such as open-source Bullet (www.bulletphysics.org) or proprietary PhysX (www.developer.nvidia.com). These physics engines provide an approximated, real time simulation of certain physical phenomena such as rigid body dynamics, soft body deformations or even fluid dynamics. Although aimed mainly at video games, these solutions were evaluated for use in simulated surgical training (Marks et al., 2007), for prototyping (Pang et al., 2010) and development (Choi et al., 2012) of surgical simulators.

When the virtual scene is defined, the engine takes care of its rendering. The final picture is then displayed on 2D or 3D screens, usually at the rates of 60Hz or 120Hz, respectively. It is accepted that, in order to achieve smooth animation, the engine has to deliver a new image at least at a rate of 60 frames per second (fps) per eye. For more demanding scientific visualization and visual data processing, specialized software such as the Visualization Toolkit (VTK, www.vtk.org) is available.

In medical applications, the input data for 3D scenes is typically human anatomy that may be obtained using medical imaging of patients. The DICOM format is widely used to exchange the 3D images acquired via CT and MRI scanners and digital databases (John, 2008). These images can be already rendered in this form (3D volume rendering) or processed using segmentation algorithms to extract the 3D polygonal surface meshes of specific organs and tissues. There are also sources of off-the-shelf 3D medical data, both commercial (e.g. www.3dscience.com) and non-commercial, most notably, The Visible Human project (Ackerman, 1998) sponsored by the U.S. National Library of Medicine (NLM).

The survey by (Vidal et al., 2006) gives a comprehensive overview of principles and applications of computer graphics in medicine.

2.4.2 Haptic interfaces

Touch, next to visual, is the second primary sensorial modality used in simulation (Coles et al., 2011). A haptic (Greek for sense of touch) device is a human–computer interface (HCI) designed to receive an input from the user and generate tactile or force feedback back to the user. A tactile response provides cues "for a fingertip" such as surface roughness, slippage or temperature, but is not yet widely used in medical applications (John, 2008). Force feedback, on the other hand, is a more mature technology and already an important component of medical simulators. Force and torque are usually generated by motors and perceived by the muscles in arms and hands. In order for this feedback to be as realistic as possible, it is accepted that the minimum haptic device update rate must be at least 0.5 kHz, and preferably 1 kHz (Coles et al., 2011). There is a range of generic commercial haptic devices available, which mainly differ on the degrees of freedom (DOFs) in which they can track the positions and/or orientations and produce force feedback. The cheapest haptic device - Novint Falcon (www.novint.com) - costs around £150 and offers 3DOF in both tracking and force feedback. A more advanced device, the Geomagic Touch (formerly Sensable PHANToM Omni, www.geomagic.com), costs around £1,300 and is able to track the attached stylus in 6 DOFs (position + orientation), generating force feedback in only 3 DOFs.

Figure 2.10: Two commercially available haptic devices: Novint Falcon (left) and Geomagic Touch.

Devices able to generate force feedback in all 6 DOFs, such as the PHANToM Premium (www.geomagic.com) or ForceDimension Delta

(www.forcedimension.com), cost upwards of £10.000. In academic applications, engineers often build custom haptic devices mimicking specific surgical instruments such as laparoscopes, catheters/guidewires (e.g. VSP, Figure 5.3) and flexible endoscopes (Figure 4.5), or modify off-the-shelf devices such as those mentioned earlier. The manufacturers of commercial simulators, such as the aforementioned Simbionix Ltd. and Mentice AB, tend to use their proprietary haptic technology.

In terms of software to control and interface with haptic devices, several higher-level haptic APIs have been developed such as Chai3D (www.chai3d.org), OpenHaptics (www.geomagic.com) or H3DAPI (www.h3dapi.org). These allow developers to add haptic rendering of a variety of (generally simple) effects to their software. However, in many cases, developers need to build their own specific haptic renderers customized to work in tandem with other simulator components.

Further details on the role of haptics in medical simulators may be found in the introduction to haptic rendering (Salisbury et al., 2004) and state of the art survey presented in (Coles et al., 2011).

2.4.3 Rigid body simulation

Creating a convincing imitation of the real-world is a non-trivial task, since real objects can interact in many different and complex ways. The simplest object to simulate on a computer using physical laws is a mass-point (also referred to as a particle). In 3D, a mass-point is defined by a vector $\mathbf{x} \in \mathbb{R}^3$ denoting its position, and a scalar mass m. A mass-point does not have an orientation. By applying Newton's laws, the equations of motion of mass-points needed for simulation of small objects such as particles of, for example, dust or water, may be derived. Connecting mass-points together using constraints (e.g. springs), enables for simulation of many interesting flexible objects such as ropes, hairs, cloths or even human organs. However, mass-points are not particularly suitable to simulate rigid bodies as it would require a large number of highly constrained points to represent the shape.

A rigid body is a solid object for which distance between every constituent pair of points will never change. In other words, it is infinitely stiff. It cannot deform or break. In contrast to mass-points, rigid bodies have an orientation that may be represented using a rotation matrix $\mathbf{R} \in \mathbb{R}^{\mathbf{3x3}}$**.** To simulate actual rotations, the distribution of mass m over body volume needs to be considered. It is captured by an inertia mass matrix $\mathbf{I} \in \mathbb{R}^{\mathbf{3x3}}$ and a centroid of mass distribution called the centre of mass (COM). Simulating a system in which bodies do not interact with each other can be easily done by solving Newton-Euler differential equations of motion. However, collisions, contacts, constraints and user interactions complicate the system. The equations of motion must be augmented to consider these effects, resulting in linear complementary problems (LCPs), which cannot be solved directly. On a computer, where the time is not continuous, but broken in small, discrete steps, solution of these equations can be approximated using specialized algorithms known as solvers. Using numerical methods, the resulting forces and torques are time integrated into new translational and angular velocities, which are in turn integrated into new positions and orientations. The choice of solver and integrator is usually a trade-off between supported functionality, accuracy and performance.

The reader is referred to a book on classical mechanics (Goldstein, 1981) for more details on rigid bodies physics. Physics-based computer simulation may be further explored in a classic Siggraph course by (Baraff and Witkin, 1997), and a recent state of the art survey can be found in (Bender et al., 2014).

2.4.4 Collision detection

Collision detection is an indispensable component of any physics simulation software. It is a classic computational problem of detecting intersecting pairs of objects at every simulation step. The algorithms used are often tailored for the specific application. For example, in physics engines aimed at games, they are optimized to efficiently handle collisions between a relatively small number of dynamic rigid bodies with large static worlds. Fast moving objects, such as bullets,

require a different approach called continuous collision detection in order not to "fly-through" other objects during discrete time steps. Game physics engines need to update the scene in sync with the visual output, which typically runs at a rate of 60Hz.

On the other hand, collision detection algorithms used in medical simulators focus on tackling a different type of problem – virtual tissues and organs should deform during interactions, therefore they need to be treated as dynamic objects. There are no fast moving bullets but, in contrast to rigid bodies, highly-deforming objects can not only collide with the surrounding environment but also with themselves, i.e. self-collide. Lastly, due to the possible interaction through haptic devices, collision detection must be fast enough to run at haptic interactive rates (0.5-1kHz).

The naïve approach, known as "brute-force", tests all objects in a virtual environment against all other objects. When considering deformable bodies, using this approach would require to test all the elements (i.e. all triangles or mass-points) of the object against the environment or, in the case of self-collisions, also with itself. This problem would have an $O(n^2)$ complexity, where n is a number of objects, which is not suitable for real-time simulation.

A common approach to tackle this problem is to quickly rule out object pairs which could not possibly collide (the broad-phase,) before doing more detailed tests (the narrow-phase). In order to speed-up calculations, objects are approximated by bounding volumes (Figure 2.11). These are usually bounding spheres, axis-aligned bounding boxes (AABBs), oriented bounding boxes (OBBs), discrete-orientation polytopes (k-DOPs) or convex hulls. The selection of the right bounding volume depends on a bounded shape and is a trade-off between efficient intersection calculations and "tighter" approximation of the bounded shape. The bounding volumes are tested for overlap before testing the bounded mesh.

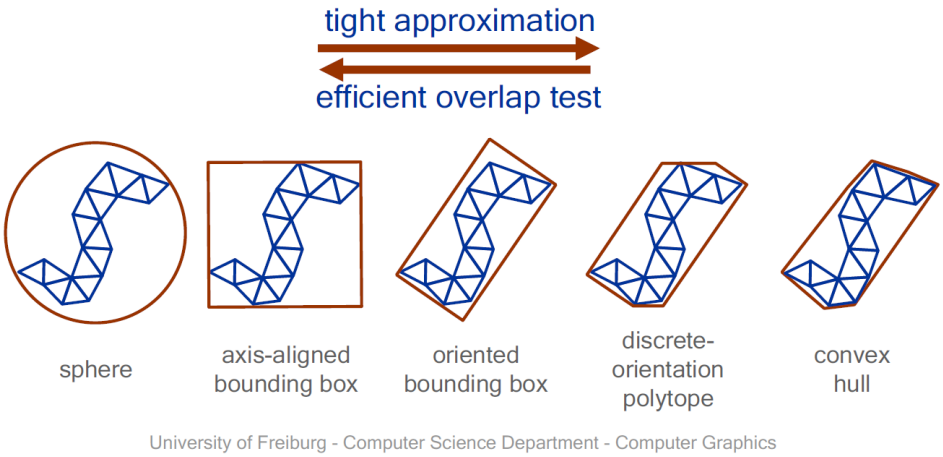


Figure 2.11: Examples of bounding volumes used in collision detection sorted by accuracy of approximation (http://cg.informatik.uni-freiburg.de).

Use of bounding volumes speeds up the intersection tests, but still requires tasting all *n* objects against all others ($O(n^2)$ complexity). Spatial partitioning is the process of dividing space into non-overlapping regions. These regions are usually organized hierarchically in tree data structures. In 3D, common methods include binary space partitioning (BSP trees), K-dimensional trees (*K*-D trees) and octagonal trees (OCTrees).

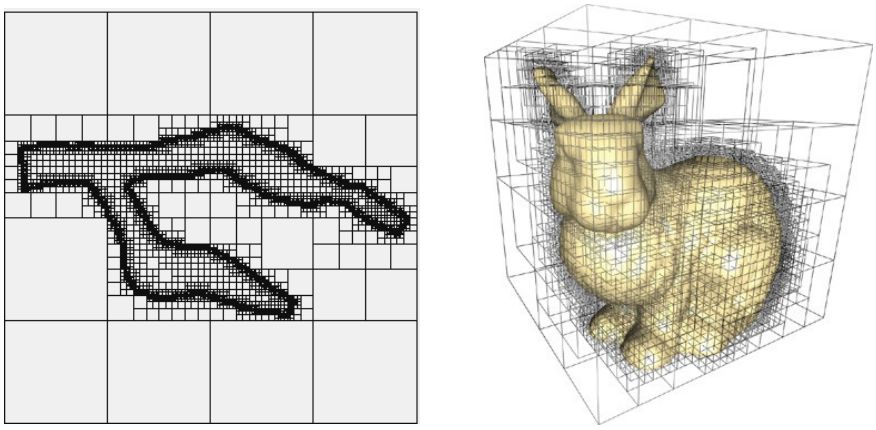


Figure 2.12: On the left: An example of 2D version of OCTree - a QuadTree. (Liang et al., 2004) On the right: The Stanford Bunny wrapped in an OCTree. (Pharr and Fernando, 2005).

A similar solution known as Bounding Volume Hierarchies (BVH, Figure 2.13) is based on bounding volumes arranged in a tree hierarchy. The main difference with

the spatial partition schemes is that two or more volumes can overlap i.e. cover the same space.

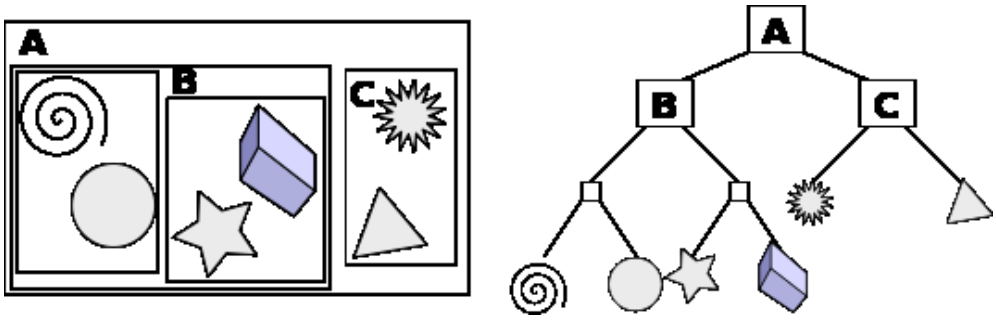


Figure 2.13: An example of a bounding volume hierarchy in 2D using rectangles as bounding volumes (http://commons.wikimedia.org).

Keeping scene objects organized in a tree structure allows for much faster geometry queries by traversing the tree from its root to its leafs pointing to actual objects. It is important to note that space partitioning and BVHs can be used not only to organize entire objects in a 3D scene, but also particular components of a 3D object. These can be polygons or triangles in a polygonal mesh, or mass-points in a mass-spring system. Figure 2.14 presents mesh polygons organized in a BVH consisting of AABBs at three different levels of detail. In fact, it is common to partition a scene using space partitioning and object geometry using BVH.

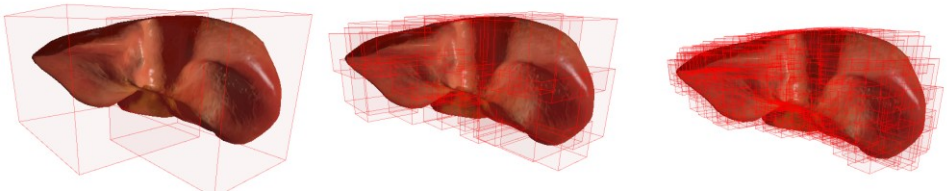


Figure 2.14: Three levels of axis-aligned bounding boxes (AABB) volume hierarchy (BVH).

The selection of optimal spatial partitioning with all its characteristics and BVH depends on a number of factors. First and foremost, on the underlying geometry and on whether objects are evenly spread in space or they concentrate in some regions. (Bergen, 1998) states that bounding volume hierarchies based on AABBs offer the best performance for collision detection of deformable bodies.

The structure of the tree can be pre-computed during the initialization phase. Construction of the tree usually follows a top-down or bottom-up approach, and considers different metrics such as maximum tree depth (height), tree degree (the number of children) or average number of triangles preferred in a leaf. A tree of a lower degree (e.g. binary trees have degree 2) will have greater depth. The number of steps to traverse the tree from root to leaf will be larger, but less work will be done at each visited node and vice versa. Usually, well-balanced trees are favoured as they offer comparable traversal times regardless of query location, thus preventing sudden performance drops. However, if objects are dynamic or deformable, the structure of the tree needs to be updated at runtime. Reconstructing the whole tree at every time-step is generally not feasible, hence trees are updated locally, which in turn may lead to a gradual degradation of tree "quality".

A review of new trends in collision detection suggested that its future lies in parallel and massively-parallel implementations (Avril et al., 2009). Indeed, a number of solutions employing Graphical Processing Units (GPUs) have emerged in the last few years (Pabst et al., 2010, Wang et al., 2012, Wong et al., 2012, Wong et al., 2014). Further details on collision detection may be found in (Bergen, 2004) and (Ericson, 2005).

2.4.5 Collision response and constraints

Collision response deals with dynamic changes of bodies following collision (e.g. deformation, displacement). The simplest form of collision response is the penalty force method first introduced in (Terzopoulos et al., 1987). When a collision occurs, a force is applied to push the objects away from it. The size of the force is proportional to the penetration depth (i.e. how much the two objects overlap) through a stiffness constant. Thus, the penalty method can be thought of as a temporary spring trying to separate colliding objects. Such simplicity is also the main disadvantage of this method, since the stiffness constant needs to be tuned to avoid large penetrations, whilst at the same time not introducing instabilities to the system. A stiffness constant value that works well in one situation, may produce

poor results when the simulation setting or environment changes slightly. Moreover, the penalty method requires that a collision and some penetration occur before the resulting forces can be computed. Despite these drawbacks, the penalty method has been successfully used in cases where simplicity and fast calculations are more important than accuracy (Martin et al., 2010)

Another way of handling collision response is by directly correcting the state of the system. One of the common approaches is impulse based (Mirtich and Canny, 1995). An impulse is a sudden change in momentum, which prevents the two objects from penetrating. The main advantage of this method over the penalty force method is that it does not require manual adjustment and does not introduce additional stiffness into the system. The objects should not overlap considerably regardless of their mass, relative velocity or simulation time-step. Applying impulses sequentially works well when bouncing two colliding objects from each other. However, it is less suitable to efficiently handle simultaneous and persistent contacts. Treating collisions and contacts in the same way may result in visible vibrations known as “jitter”. Jitter is particularly visible when the objects are stacked one on top of another, as an impulse applied to resolve one contact will affect all other impulses generated to solve other resting contacts. To address this, contact impulses are often handled differently using a system of linear equations. Despite impulses not providing physically accurate results, they are a simple and robust method for resolving interactions between rigid bodies in a visually plausible way (Mirtich and Canny, 1995).

Another approach is called Position Based Dynamics (Muller et al., 2007). It simulates dynamics using the Verlet integration (Verlet, 1967, Jakobsen, 2001) – a method widespread in molecular dynamics simulation. It solves a system of non-linear constraints using a Gauss-Seidel iteration by directly updating positions of the mass-points. It derives momentum changes implicitly from the position up-dates, thus avoiding typical instabilities associated with explicit integration. This method was later applied to fluid simulation (Macklin and Muller, 2013) and, more recently, in a unified parallel framework for particle physics (Macklin, 2014) (Figure

2.15). In this approach, particles connected by various constraints are employed to simulate deformable bodies, cloths, liquids, gases and even rigid bodies including bilateral interactions between all of them.

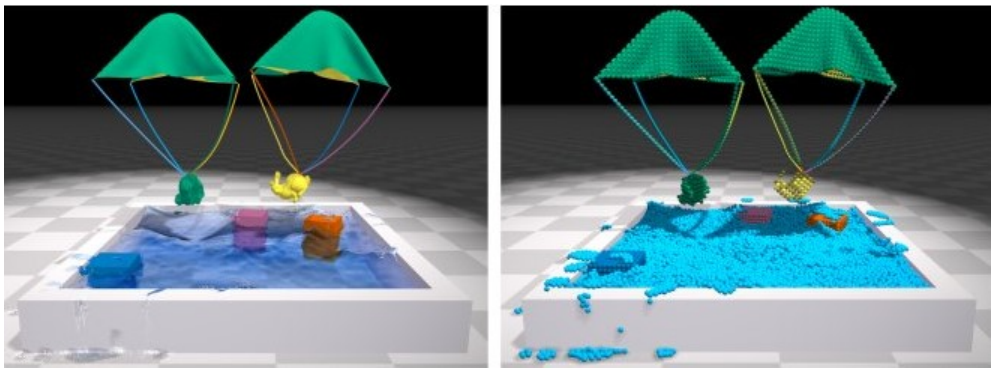


Figure 2.15: NVidia FLEX - a unified particle solver based on Position Based Dynamics. (Macklin, 2014).

Constraint based methods aim to find an exact solution by augmenting the equations of motion. This results in the complementary problems (LCPs) mentioned earlier. Constraint based methods are mathematically correct since they are based on Newtonian dynamics, but much harder to implement and can be computationally too expensive for interactive simulations. However, this approach can produce more accurate results and be efficient when simulating complex physical systems with many constraints involved.

The approaches to solving the LCP problem are exact direct global methods (Baraff, 1994, Eberly, 2010), local iterative methods based on a Gauss-Seidel method of solving a system of linear equations (Catto, 2005, Erleben, 2007) or a combination of these two called block iterative solvers. All are discussed in more depth in the context of elastic rods simulation in Section 3.1.2 of Chapter 3.

2.4.6 Deformable body modelling

Human tissue, due to its non-linearity, anisotropy and viscoelasticity, is one of the most challenging deformable bodies to model and simulate. Mathematical models developed in disciplines such as mechanical engineering or material sciences focus primarily on achieving the highest possible accuracy, regardless of computational time. On the other hand, visual plausibility and real-time (on-line) interactivity are

more important in computer graphics. Surgical simulation sits between these two disciplines. It requires models which are not only visually convincing, but also “feel” right to the user during interactions via a haptic device.

Mass-spring system (MSS) is a popular physics-based model. Simulating a deformable model using MSS requires discretising the 3D mesh of an object into mass-points (nodes) connected by weightless springs. Dampers are often added to improve stability. During each physics update (time-step), the forces exerted by the springs on each mass-point are calculated using a form of Hooke’s law, and later integrated into new velocities and positions. MSS is fast, intuitive and easy to implement. Such system can handle even large deformations with relative ease and is well suited for parallel processing. However, simple MSM implementations are a significant approximation to the continuous body, which results in a number of drawbacks. First, model behaviour is dependent on its geometry and topology. Change in discretization requires retuning parameters of springs and dampers. Second, it is difficult to derive the parameters from the physical model. MSSs are usually tuned manually to behave as the reference object, which can be a very tedious process. While using a fast explicit integrator, the local nature of deformations causes delays in force propagation through the system and difficulties with volume preservation. Finally, it is not suitable for simulation of stiffer bodies as this would require small time-steps to keep the system stable.

Several attempts have been made to improve the MSS approach and address the above problems, with the majority being application specific. (Provot, 1995) added a correction step after integration to prevent model overstretching and simulate non-linear behaviour in cloth simulation. (Bourguignon and Cani, 2000) extended the model by adding anisotropic behaviour and volume preservation. More recently, (Lin et al., 2010) presented a heterogeneous model for simulating bio-tissues.

An alternative approach addressing the problems of MSS related to deformation propagation and volume preservation was introduced in (Frisken-Gibson, 1997).

The neighbouring elements of an object are linked together like elements of a chain mail, hence the name – 3D ChainMail. The algorithm consists of two stages: deformation propagation and elastic relaxation. During the first phase, each element can move freely within certain geometric limits, without influencing its neighbours, while major displacements are propagated immediately to the adjacent elements. The second process iteratively reduces the system energy and its more advanced versions can even recreate non-linear elastic and plastic materials. The 3D ChainMail has been applied to surgical simulation, yet its use is not very widespread (Meier et al., 2005). While improving the volumetric nature and enabling for complex bio-mechanical behaviour in comparison to simpler MSS implementations, it is still-far-from-accurate volumetric behaviour.

A more physically accurate approach directly based on the laws of continuum mechanics is the Finite Element Method - FEM (Zienkiewicz and Taylor, 2000). In FEM, the deformable object is modelled as an elastic continuum by partitioning it into a large number of small elements, typically tetra- or hexahedral, whose dynamics are governed by partial differential equations (PDEs). Linear FEM models can accurately recreate only small deformations in linear complexity O(n) with respect to number of elements. . However, applications of non-linear FEM, which can recreate even complex deformations, are limited due to large computational complexity. Much effort has been devoted to speeding up FEM in order to make it suitable for real-time simulations. Some of the first experiments with FEM in real-time surgical simulation were those by (Bro-Nielsen, 1998) and (Cotin et al., 1999), who used several pre-processing methods to enable real-time interaction. (Berkley et al., 1999) used banded matrices to speed-up calculations and later applied FEM to a virtual suturing simulator (Berkley et al., 2004). (Nesme et al., 2006) applied a hierarchical FE technique to control the resolution of the models, thus allowing to concentrate the computations in the region of user interaction.

Despite these attempts, FEM was until recently still considered as challenging for real-time simulation (Moore and Molloy, 2007). However, in the past few years, the FEM techniques were significantly accelerated using massively-parallel

computations implemented on graphics processing units (GPUs). For example, (Comas et al., 2008) presented an efficient non-linear FEM model of soft tissues. (Courtecuisse et al., 2011) applied a GPU-based FEM for interactive simulation of liver resection. (Allard et al., 2012) describes methods to implement an implicit finite element solver on the GPU in a *"GPU Computing Gems"* book by (Hwu, 2012).

As stated in an excellent survey on this matter *"None of the deformable models presented above simultaneously exhibits all of the sought-after characteristics required in surgery simulation such as speed, robustness, physiological realism, and topological flexibility."* (Meier et al., 2005). A more recent review can be found in (Cueto and Chinesta, 2014).

2.4.7 One-dimensional deformable bodies and elastic rods

One-dimensional deformable bodies are objects having one dimension, namely "the length", much larger than its cross-section. They can be used to model threads, ropes, wires, cables, etc, and may be visually approximated as smooth curves in space. Elastic rods are one-dimensional deformable bodies characterized by having large non-linear deformations even if the local strains are small. The behaviour of elastic rods is governed by the theory of elasticity – "the elastica" – developed in the 18th century by Leonhard Euler (1707 - 1783) and Jakob Bernoulli (1655 - 1705). The elastica was later generalized *by the Cosserat Theory (Cosserat and Cosserat, 1909)*, which is considered as a final step in the formulation of a modern theory of elastic rods (Goss, 2003). The Cosserat Theory was formulated in the beginning of 20th century by two French brothers, Eugène and François. Eugène (1866 – 1931) was a mathematician, chair of the Astronomy Science Faculty at Toulouse University. His brother, François (1852-1914), graduate from École Polytechnique, was a civil engineer. Together, they were led into elasticity theory through a study of surface mechanics. They proposed modelling a surface as a mesh of curves with intrinsic directions at each point. In other words, each point lying on a curve had its own local coordinate system (a director) which was moving together with them when the surface was undergoing a deformation. By comparing the directors before and after the deformation, the change in the surface could be specified. This

approach was later extended to elastic rods. A comprehensive theory on problems of elasticity is given in (Antman, 1995).

Elastic rods may undergo stretching, bending and twisting deformations. Rods that can be easily stretched or compressed are known as “extensible rods”, whereas “inextensible rods” demonstrate high resistance to changes in length. Bending is the deformation of the rod centreline resulting in a bending moment that works against it. The rods can have an intrinsic bending resulting in a resting shape different than a straight line, for example, a curve or a spiral. Torsion is a twist around the centreline causing a torque working against it. Efficient and realistic torsion recreation is considered as the main challenge in the modelling of elastic rods (Spillmann, 2008).

Modelling and simulation of elastic rods has been an active area of research. In mechanical engineering, where it is common to provide the starting and end points of a rod, the approaches emphasise solving a resulting boundary value problem (Pai, 2002, van der Heijden et al., 2003). In computer graphics, elastic rods have been used to model ropes and hairs for physics-based animation. Several real-time models of elastic rods have been used in the field of medical simulation to simulate sutures or flexible surgical tools such as guidewires and catheters.

The simplest techniques to model one-dimensional deformable bodies are based on the aforementioned mass-spring model. The rod is discretized into a number of mass-points connected by springs forming a chain. In one of the earliest approaches proposed for hair modelling (Rosenblum et al., 1991), linear springs maintain the given length between the neighbouring mass-points and angular springs behave as hinges trying to preserve the given angle as shown in Figure 2.16. The penalty method is used for collision response and explicit Euler integration for time-stepping. This model shares all the pros and cons of MSS. Namely, it is straightforward to implement, but hard to tune and making the rod even visually inextensible would result in an additional stiffness requiring smaller time-steps to integrate.

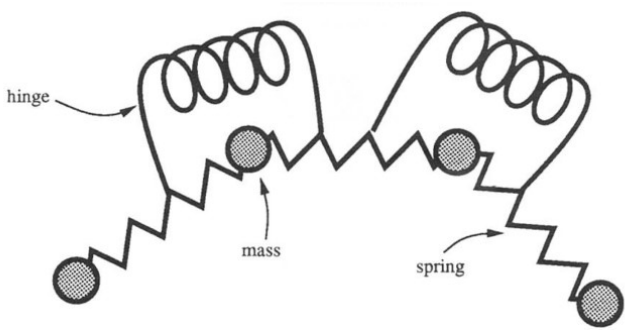


Figure 2.16: Simple mass-spring model for elastic rods modelling. After (Rosenblum et al., 1991)

Around the same time, (Anjyo et al., 1992) presented another approach for hair simulation based on a simplified cantilever beam. By neglecting shear deflection, they derived simple differential equations that can be analytically solved and prevent hair from stretching. Both solutions did not account for torsional deformations. This was addressed much later by (Selle et al., 2008) who introduced a MSS of tetrahedral "altitude springs" to recreate the hair twist, thus enabling the simulation of curly hair. (Ward et al., 2007) presents a review of simulation and modelling of hair. The reader is also referred to the Siggraph'06 course notes by (Hadap, 2006). More recently, (Casati and Bertails-Descoubes, 2013) developed a clothoid-based model to resolve the hair dynamics with very few control points and (Chai et al., 2014) proposed a reduced coordinates formulation for interactive hairs.

Mass-spring systems have also been applied to the simulation of knot tying. (Wang et al., 2005a, Wang et al., 2005b) introduced "torsion springs" expressed with a scalar angle for suture simulation. (Phillips et al., 2002) presented an adaptive model where the number of mass-points was dynamically increased in sections where knots appear. To ensure rope inextensibility, (Brown et al., 2004) proposed a fast approach based on geometry rather than physics, named *Follow the Leader*. This was later improved and applied to dynamic hair simulation by (Müller, 2012) using their Position Based Dynamics methods.

A different method that can recreate torsion, as well as inextensibility, employs rigid multi-body serial chains borrowed from robotics (Featherstone, 1987). (Hadap, 2006) used differential algebraic equations (DAEs) to solve the resulting

system. However, due to hard constraints linking the rigid bodies, the equations are stiff requiring complex techniques to solve them. To tackle this, (Choe et al., 2005) presented a hybrid solution that connected the rigid bodies in a chain not by hard constraints, but by linear and angular springs instead.

The approaches by (Choe et al., 2005) and (Hadap, 2006) have an important advantage over previous ones. Namely, by having explicit orientations of the rigid bodies, they can handle both bending and torsion deformations in the same way. Therefore, the bending deformation can be balanced out by the twist deformation, and vice versa. This results in a desirable looping phenomenon.

(Pai, 2002) introduced a solution based on the Cosserat rod theory. In order to simulate a strand of surgical suture, he derived a set of spatial ordinary differential equations (ODEs) that can be efficiently integrated in two passes. However, his solution does not explicitly simulate the centreline, but reconstructs it using the specified position and orientation of start and end points, which complicates collision handling (Spillmann and Teschner, 2007).

(Bertails, 2005) improved Pai's work by using energy minimization to compute the equilibrium position of strands for hair simulation. This allowed for handling external forces such as gravity and collision response, but was static, thus preventing its use in animation. In her following paper, (Bertails et al., 2006) simulated the dynamics of Cosserat rods using Lagrangian mechanics. However, the computation time was quadratic in respect to the number of elements, preventing its application in accurate interactive simulation.

Later, (Gregoire and Schomer, 2007) presented a solution also based on Cosserat theory in which they place a quaternion governing the material frame orientation between the neighbouring mass-points. However, this results in the centreline (the mass-points positions) becoming separated from the material frames (the quaternions). The centreline and the material frames need to be explicitly adapted to each other. This can be achieved, for example, by a constraint enforced by a penalty method as in (Gregoire and Schomer, 2007) or Lagrange multipliers as in

(Spillmann and Harders, 2010). Some authors argue that such an explicit representation of the centreline facilitates the simulation of contacts and looping phenomena (Spillmann and Teschner, 2007, Bergou et al., 2008).

Other approaches to Cosserat Rod include (Theetten et al., 2008) who combined a Cosserat approach with a geometric spline-based deformation model and employed the spline control points as degrees of freedom of the rod. (Bergou et al., 2008) solved dynamic deformations using a discrete differential geometry formulation. They model the twist of the material as a deviation from a canonical frame and present a method to evolve this quasi-statically. (Kmoch et al., 2009) tailored this model to real-time hair simulation and later presented a massively-parallel implementation running on the GPU (Kmoch et al., 2010). Recently, a solution integrating the Cosserat theory with the aforementioned Position Based Dynamics (Muller et al., 2007) was presented (Umetani et al., 2014). This is especially interesting in the context of GPU-based unified physics solvers such us NVidia FLEX (Macklin, 2014) or Autodesk Nucleus (Stam, 2009).

A few other approaches to elastic rods in the context of cardiovascular interventions, which are most often adaptations of the above models to simulate catheters and/or guidewires, is given in Section 5.2.

2.4.8 Simulation frameworks

As stated in the previous sections, medical simulation is a complex multi-disciplinary field combining visualization, haptic rendering, physics-based simulation and collision detection. In order to allow researchers to focus on developing new, cutting-edge algorithms and avoid “reinventing the wheel”, the need for suitable development tools has appeared.

Over the recent years, several open source simulation frameworks have been developed such as SPRING (Montgomery et al., 2002), GIPSI (Cavusoglu et al., 2004) or VRASS (Kuroda, 2008). One of the most recent open-source frameworks, which are still actively developed, are SOFA (Allard et al., 2007) and OpenSurgSim (*www.opensurgsim.org*). SOFA was created by research groups from CIMIT and

INRIA. Its architecture relies on multi-model representations (visualization, deformation, collision) of an object which are mapped together and organized in a scene-graph. SOFA offers a range of built-in algorithms, the possibility to add new ones and combine it all together. OpenSurgSim is maintained by SimQuest Solutions Inc. (www.simquest.com) and sponsored by the US Army Telemedicine & Advanced Technology Research Center. It offers an open framework that includes the necessary building blocks for surgical simulations, such as native device support, haptic feedback, graphics, discrete collision detection and physics simulation.

A recent survey on software development tools for surgical simulation (Ruthenbeck and Reynolds, 2013) concludes that, although the aforementioned solutions and frameworks can indeed contribute to simulator development, the perfect tool does not exist and combining a range of existing techniques is challenging.

2.5 SUMMARY

In the first part of this chapter the reader was introduced to the concept of simulation based-training by a brief historical sketch summarizing the most notable milestones in the development and application of simulation for aviation, entertainment and, finally, in medicine and surgery. This part finishes with the presentation of virtual reality simulators whose pros and cons have been outlined in the previous chapter in Section 1.1.3.

The second part of this chapter reviews both software and hardware components of a modern VR surgical simulator. It describes the current state-of-the-art methods for real-time 3D graphics, haptic interfaces and physically-based simulation, including rigid and deformable bodies modelling and simulation, as well as techniques for collision detection and response. The results of this review establish the technical fundamentals and design choices upon which more specific solutions will be built and research carried out in the following chapters. Substantial attention was given to one-dimensional deformable bodies, a.k.a. elastic rods. The implementation of such a mathematical model is a starting point for the simulation of flexible surgical tools and is presented in the next chapter.

Chapter 3

MODELLING AND SIMULATION OF ELASTIC RODS

In its second part, Chapter 2 gave an overview of the previous work in the modelling of elastic rods. This chapter presents the model of choice used as a foundation for virtual flexible surgical instruments – the CoRdE model (Spillmann and Teschner, 2007). The CoRdE model is based on the Cosserat theory of elastic rods introduced in the previous chapter. A modification to the CoRdE enabling efficient enforcement of rod inextensibility is proposed.

Section 3.1.1 gives a continuous formulation of the Cosserat rod followed by the method for its discretization used in the CoRdE implementation. Section 3.1.2 elaborates on the constraints used in the original model and proposes a modification ensuring inextensibility and incompressibility. Section 3.1.4 focuses on an approach used to detect and resolve collisions and self-collisions of the virtual rods.

The results presented in Section 3.2.1 concentrate on the computational performance of the implementation in comparison to other equivalent or similar approaches, whilst 3.2.2 examines rod inextensibility. Lastly, Section 3.2.3 investigates the behaviour of the rod and collision scheme in an interactive case during which the two ends of the rod are controlled by a pair of haptic devices. A chapter summary is given in section 3.3.

3.1 METHODS

The Cosserat theory of elastic rods introduced in Chapter 1 assumes that each point lying on a rod centreline has its own local coordinate system – a director – which is moving together with this point when the rod is undergoing a deformation (Figure 3.1). By comparing the directors, the stretch, bend and twist deformations of the rod can be quantified.

The CoRdE model by (Spillmann and Teschner, 2007) is a method of choice for the underlying physically-based model of flexible surgical tools. First, because it is based on Cosserat theory – a solid theoretical foundation, which is considered as a final step in the formulation of a modern theory of elastic rods (Goss, 2003). Second, because CoRdE is a fast, dynamic and elegant solution, which is also reasonable in terms of implementation complexity.

The CoRdE uses an explicit centreline representation to model material stretch and material frames adapted to the centreline by penalty forces to recreate the bending and twisting deformations. Some authors argue that such an explicit representation facilitates the simulation of contacts and looping phenomena (Spillmann and Teschner, 2007, Bergou et al., 2008). It also simplifies the overall implementation, internal friction calculations and visualization.

However, the original CoRdE rod is extensible and compressible. This is not desirable in the context of surgical tools. Flexible endoscopes, catheters or guidewires practically do not change their length. The rod stretch and compression can result in a perceptible latency to user manipulations at the proximal end of the rod, especially in case of longer rods. Due to the penalty method used in CoRdE, the reduction of this effect would introduce an additional stiffness to the system. As a result, the stable simulation would require smaller time-steps which, in turn, would degrade the performance. To tackle this, in Section 3.2.1 the CoRde model is modified to make it inextensible and incompressible, while maintaining the computational efficiency of the original model.

The following formulation is based on the work by (Spillmann and Teschner, 2007) and (Spillmann and Harders, 2010), followed by the explanation of the aforementioned modification.

3.1.1 Continuous formulation

The centreline of the rod is represented by a function mapping line parameter ***s*** to a position in 3D space $\mathbf{r} = \mathbf{r}(s): [\mathbf{0}, \mathbf{1}] \rightarrow \mathbb{R}^{\mathbf{3}}$. To represent the stretching deformation along it, the strain vector $\mathbf{v} = (v_1,\ v_2,\ v_3)^{\boldsymbol{T}}$ is defined**,** which is a rate of change in the position of the centreline. By assuming that the rod is unshearable $v_1 = v_2 = \mathbf{0}$ **,** the stretch along the centreline is equal to $\boldsymbol{v_3} = ||\mathbf{r}'||$ **.** An unstretched rod has $v_3 = \mathbf{1}$.

In order to represent bending and twisting deformations, the concept of material frames is introduced. The material frame is an orthonormal basis $\mathbf{d}_k$, k = 1, 2, 3, where $\mathbf{d}_k$ are called directors. The first and second director indicate the orientation of the centreline, whereas the third one, $\mathbf{d_3}(s)$, is always adapted to the curve, i.e. parallel to the tangent $\mathbf{r'}(s)$ at the same point (Figure 3.1). From the directors, the rotation matrix $\mathbf{R}(s) \in \mathbb{R}^{\mathbf{3x3}}$ can be derived**.**

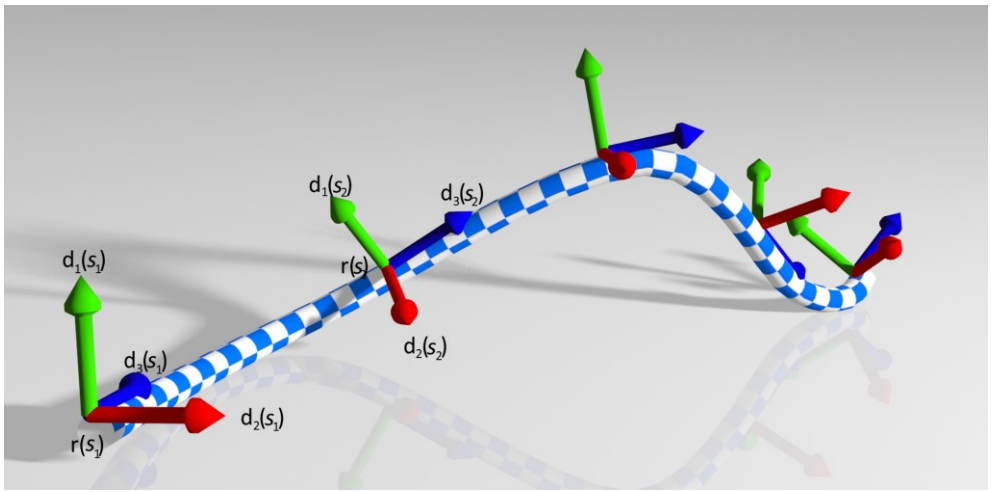


Figure 3.1: The material frames adapted to the rod's centreline.

Using differential geometry, from the material frames an orientational rate of change in the reference frame – the *Darboux vector* – $\mathbf{u_0} \in \mathbb{R}^{\mathbf{3}}$ may be obtained. Its

components are the areas swept by the directors when proceeding from *s* to (*s* + Δ*s*).

$$\mathrm{u}_0(s) = \frac{1}{2}\sum_{k=1}^{3} \mathrm{d}_k(s) \times \mathrm{d}_k'(s) \tag{1}$$

$\mathbf{d}_k'$ is a partial derivative of the material frame with respect to the line parameter *s*. After rotating $\mathbf{u_0}$ from the reference frame into the local frame $\mathbf{u} = \mathbf{R^T u_0}$, $\mathbf{u}$ relates to bending and twisting strains (Figure 3.2).

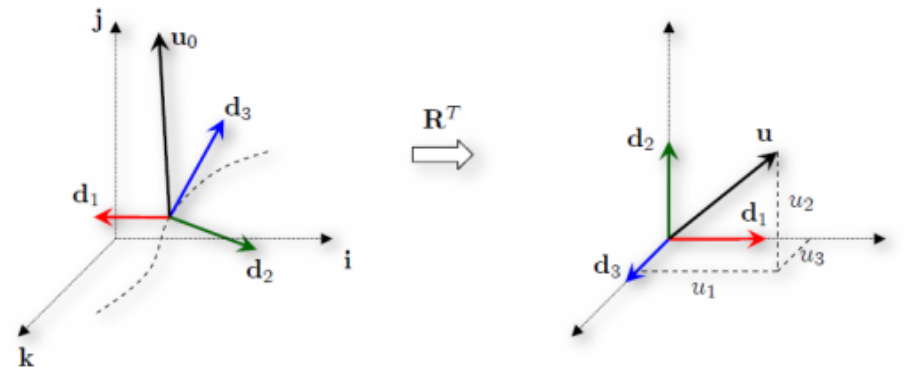


Figure 3.2: Darboux vector. After (Spillmann and Teschner, 2007).

Based on the defined strain rates and assuming a linear strain-stress relationship, the stretch energy of the entire rod can be derived:

$$V_s = \frac{1}{2}\int_0^1 K_s(v_3 - 1)^2 \mathrm{d}s \tag{2}$$

Where $K_s = E_s \pi r^2$ is a stiffness constant, E_s is a stretching Young's modulus and *r* is the radius of the rod's cross-section. The bending energy is calculated respectively:

$$V_b = \frac{1}{2}\int_0^1 \sum_{k=1}^{3} K_k(u_k - \hat{u}_k)^2 \,\mathrm{d}s \tag{3}$$

Where $K_1 = K_2 = E_b \frac{\pi r^2}{4}, K_3 = G\frac{\pi r^2}{2}$, E_b and *G* are Young's and shear modulus governing the bending and torsional resistance, and $\hat{u}_k$ are the intrinsic bend and

twist parameters. They are used to control the resting shape of the rod, for example, to model curved rods. By minimizing $V = V_s + V_b$ and by treating bending and torsion in a unified manner, the strain rates may be coupled together. The twist deformation is balanced out by the bend deformation and vice versa, which can result in the looping phenomenon.

3.1.1.1 Discretization

The centreline of the rod is discretized into *N* mass (control) points $\mathbf{r_i} \in \mathbb{R}^3, \boldsymbol{i} \in [\mathbf{1}, \boldsymbol{N}]$ and *N* – 1 material frames $\mathbf{R_j} \in \mathbb{R}^{3x3}$ as shown in Figure 3.3:

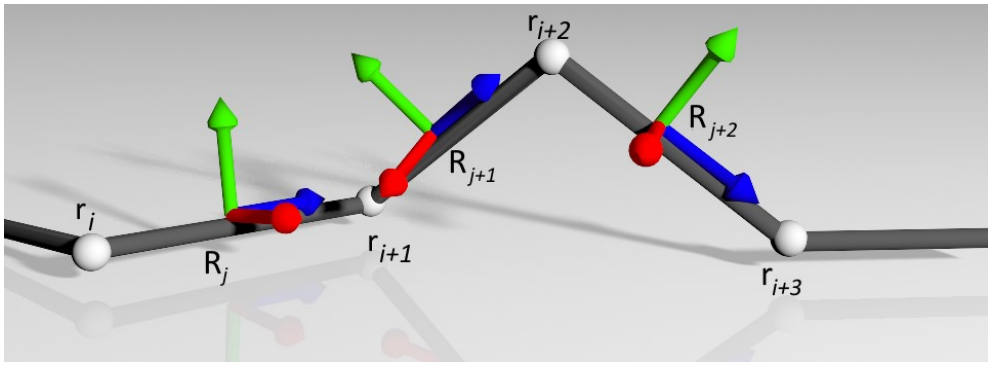


Figure 3.3: Discretized centreline of the rod with material frames.

The spatial derivative of the centreline is approximately:

$$\mathrm{r}_i' \approx \frac{\mathrm{r}_{i+1} - \mathrm{r}_i}{||\mathrm{r}_{i+1} - \mathrm{r}_i||} \tag{4}$$

In order to provide a singularity-free parameterization, the material frames $\mathbf{R}_\mathrm{j}$ are expressed by unit quaternions $\mathbf{q}_j$. Thus, the spatial derivative is approximated as:

$$\mathrm{q}_j' = \frac{1}{l_j}(\mathrm{q}_{j+1} - \mathrm{q}_j) \tag{5}$$

where l_j is the step length. Using quaternions enables to fully determine the material frame without considering the centerline points. The actual strain rates can be computed as:

$$u_k = 2\mathrm{B}_k\mathrm{q} \cdot \mathrm{q}' \tag{6}$$

where $\mathbf{B_k} \in \mathbb{R}^{4x4}$, ***k* = 1, 2, 3** are constant skew-symmetric matrices. The detailed derivation of $\mathbf{B_k}$ is given in the appendix of (Spillmann and Teschner, 2007). The CoRdE model also considers internal friction forces which damp relative motion in the rod. They are modelled as additional dissipation energy terms, both translational and angular.

3.1.2 Constraints

The mass-points and material frames of the elastic rod are subject to a number of restrictions constraining its movement. Constraints can be defined by specifying legal positions (position constraint) of a state vector $\mathbf{x}$ and can be mathematically expressed by a scalar function $\mathbf{C}(\mathbf{x})$. For the Cosserat Rod, four types of constraints are specified in the proposed implementation: parallel, unit quaternion, distance and collision.

The bending energy as defined above depends solely on the configuration of the material frames. In order to couple the material frames with the centreline, a parallel constraint C_p is defined, which aligns the third director $\mathbf{d}_3$ of the material frame with the tangent $\mathbf{r}'$ of the centreline.

$$C_p = \mathrm{r}' - \mathrm{d}_3 = 0 \tag{7}$$

To maintain the above constraint, the penalty method is used resulting in the following penalty energy equation:

$$E_p = \frac{1}{2}\int_0^1 K_p \left(\frac{\mathrm{r}'}{\|\mathrm{r}'\|} - \mathrm{d}_3\right) \cdot \left(\frac{\mathrm{r}'}{\|\mathrm{r}'\|} - \mathrm{d}_3\right) \mathrm{d}s \tag{8}$$

where K_p is a numerical spring constant that depends on the simulated material, and $\|\mathbf{r}'\|$ is the element length. The advantage of the penalty method is its simplicity. It acts as an additional force on the mass-points and torque on material frames and it can be enforced locally, which is important for parallel implementation. The disadvantage is a possible loss of accuracy, as the constraint may not be always exactly satisfied. In addition, the higher values of K_p necessary for the simulation of stiffer materials require a smaller time step. In (Spillmann and

Harders, 2010), the authors eliminate the penalty method by using more accurate, but less efficient and global (i.e. harder to parallelize) Lagrange multipliers.

Only unit quaternions represent rotations. Thus a unit quaternion constraint may be defined as follows:

$$C_q = \|\mathrm{q}\| - 1 = 0 \tag{9}$$

The C_q constraints are enforced using coordinate projection, i.e. by renormalizing quaternions at each simulation step.

In many applications having an extensible rod is not desirable. Reducing its stretching by increasing the K_s constant to higher values would also require decreasing the time-step. However, assuming rod inextensibility, it is possible to omit calculating V_s and replace it with a distance constraint C_d, which will try to maintain the desired rest length l between the centreline's mass-points:

$$C_d = \|\mathrm{x_i} - \mathrm{x_{i+1}}\| - l = 0 \tag{10}$$

C_d is an equality constraint as all legal positions of **x** are those that satisfy $\mathbf{C}(\mathbf{x}) = \mathbf{0}$.

Contact constraints C_c are responsible for handling collisions and self-collisions with Coulombian friction. They prevent the rod from penetrating through the meshes, itself or other rods. Contrary to the previous constraints, the collision constraint is an example of an inequality constraint $\mathbf{C}(\mathbf{x}) \geq \mathbf{0}$. In other words, this constraint is imposed only when the two objects are penetrating. Omitting this condition would result in the "gluing" of objects together rather than separating them.

3.1.2.1 Constraints enforcement

Four types of constraints were defined above. The parallel constraint is enforced at the force level and the unity quaternion constraint is a simple normalization. In order to employ distance and collision constraints, Lagrange multipliers are

employed using the $\mathbf{JM^{-1}J^T}$ projection. This approach augments the equations of motion by introducing a constraint force $\mathbf{f_c}$:

$$\ddot{\mathbf{x}} = \mathbf{W}(\mathbf{f} + \mathbf{f_c}) \tag{11}$$

where **x** is a global coordinate vector gathering all the positional degrees of freedom of the rod, **W** is an inverse of mass-matrix and **f** is a global force vector**.** $\mathbf{f_c}$ annihilates the accelerations violating the constraints and is derived by combining time derivatives of positional constraint functions and *the principle of virtual work.* Specifically, a time derivative of **C** yields the velocity constraint function (12). Since **C** is a function of positions which are themselves functions of time, a chain rule may be performed:

$$\dot{\mathrm{C}} = \frac{\partial \mathrm{C}}{\partial \mathrm{x}} \dot{\mathrm{x}} \tag{12}$$

The matrix $\frac{\partial \mathbf{C}}{\partial \mathbf{x}}$ is called *the Jacobian* of **C** and is denoted by **J**. The *Jacobian* rows contain the gradients (highest rates of increase) of the scalar components of the constraint function ***C***, i.e. vectors pointing in the direction of the illegal movement. $\dot{\mathbf{x}}$ is a vector of velocities denoted onward as **v**:

$$\dot{\mathrm{C}} = \mathrm{Jv} \tag{13}$$

Differentiating again with respect to time yields the acceleration constraint:

$$\ddot{\mathrm{C}} = \dot{\mathrm{J}}\mathrm{v} + \mathrm{J}\dot{\mathrm{v}} \tag{14}$$

If it is assumed that the initial positions **x** and velocities **v** are legal, the task is to find a constraint force, which added to applied forces, guarantees that $\ddot{\mathbf{C}} = \mathbf{0}$**.** After re-arranging to match the form of **Ay = b** and introducing *the principle of virtual work,* which ensures that constraints will not change the energy of the system, i.e. do any work, (15) is obtained. Please refer to (Baraff and Witkin, 1997) for further details.

$$\mathrm{JWJ^T}\lambda = -\dot{\mathrm{J}}\dot{\mathrm{x}} - \mathrm{JWf} \tag{15}$$

$\boldsymbol{\lambda}$ is a *Lagrange multiplier*, a vector of undetermined signed magnitudes of the constraint forces for which to solve. However, numerical solutions to ODEs are subject to drift. To prevent this drift from accumulating, an extra feedback term is added which can be incorporated into the constraint force calculations. Instead of solving for $\ddot{\mathbf{C}} = \mathbf{0}$, it is solved for $\ddot{\mathbf{C}} = -k_s\mathbf{C} - k_d\dot{\mathbf{C}}$, where k_s and k_d are spring and damping constants. This feedback term is called *the Baumgarte stabilization term* and acts like a spring that pulls back the mass-points to a valid state if drift occurs. Therefore, the final equation with feedback is:

$$\mathrm{JWJ^T}\lambda = -\dot{\mathrm{J}}\dot{\mathrm{x}} - \mathrm{JWf} - k_s\mathrm{C} - k_d\dot{\mathrm{C}} \tag{16}$$

Equation (16) is now in the form of **Ay = b** where $\mathbf{A} = \mathbf{JWJ^T}$, $\mathbf{y} = \boldsymbol{\lambda}$ and $\mathbf{b} = -\dot{\mathbf{J}}\dot{\mathbf{x}} - \mathbf{JWf} - k_s\mathbf{C} - k_d\dot{\mathbf{C}}$. It enables solving for $\boldsymbol{\lambda}$, which multiplied by $\mathbf{J^T}$ will yield the constraint force vector $\mathbf{f_c}$.

Acceleration-level constraints discussed so far are not particularly well-suited for dealing with impacts during collisions or cases where, for example, constraints reach their limit angle. This requires instantaneous change of objects velocities, which in turn implies applying large forces that may result in instabilities or even break the simulation. That is why modern physics engines usually compute constraints at the velocity-level and apply them via impulsive forces. Moreover, velocity-level constraints are easier to derive and faster to compute as they do not require taking the second derivative of the constraint equation $\ddot{\mathbf{C}}$. The Equation (17) shows the system for solving for constraint impulse magnitude at the velocity-level:

$$\mathrm{JWJ^T}\lambda = -\mathrm{JV} - k_s\mathrm{C} \tag{17}$$

However, by simply solving Equations 16 or 17 and applying the resulting constraint forces or impulses, it is only possible to handle equality constraints. As mentioned before, the collision constraints during the contact will 'glue' the colliding bodies together, instead of keeping them separated. As a result, an additional inequality constraint is needed that is only active if the bodies are penetrating each other i.e. $\mathbf{C} \leq \mathbf{0}$. Adding such an inequality term into the system results in a - a *linear*

complementarity problem - LCP (Cottle et al., 2009). There are two main approaches to solving such a problem.

3.1.2.2 Exact global methods

Direct global methods, also referred as *pivoting methods*, compute the exact solution in a finite number of steps by recursive solution of systems of linear equations using a specialized LCP solver. The most popular approaches are Dantzig's algorithm adapted for rigid body simulation and extended by friction handling in (Baraff, 1994), and Lemke's algorithm described in (Eberly, 2010). Implementing an efficient and robust LCP solver is a non-trivial task as numerical errors may lead to incorrect results or even prevent from finding a solution at all. Moreover, using exact global methods can become infeasible when adding contact and friction for more than a few hundred bodies (Bender et al., 2014). The exact solvers can be highly specialized, for example, to solve a system of linked bodies in generalized coordinates using Featherstone's Articulated Body Method (Featherstone, 1987, Kokkevis, 2004). A range of numerical methods for solving the LCPs can be found in (Erleben, 2013).

3.1.2.3 Iterative local methods

Another approach are local iterative methods. The fastest and most robust techniques currently used (Catto, 2005, Erleben, 2007) are based on a Gauss-Seidel method for solving a system of linear equations (Golub and Van Loan, 2013).

In a Gauss-Seidel solver the matrix rows in a system in Equations 16 or 17 governing contacts and constraints are modelled in a unified way as equality constraints and are solved locally, one by one. By solving one constraint at a time, it is likely that it will violate other constraints. However, by repeating this process multiple times, the solver will eventually converge to the global solution in linear time. Furthermore, even if the solver is interrupted earlier, an intermediate result can be sufficient enough for the simulation to proceed. Inequality constraints for collisions and some constraint types are handled by simply projecting (or clamping) the solution at each iteration within lower and upper limits (*Projective Gauss-Seidel*).

It is clear that the number of solver iterations is critical for the accuracy of the simulation. Insufficient number of iterations may result in visible constraints drift. The most problematic cases for the iterative solvers are large stacks of bodies, especially when the mass ratios between them are large, for example, a heavy box resting on top of much lighter or heavy tank on tracks. This problem was mitigated by considering temporal coherence (Catto, 2005) or using shock-propagation (Erleben, 2007). In medical simulation large mass ratios do not occur. However, the number of solver iterations has a significant impact on deformation propagation. For example, a long flexible surgical tool such as a guidewire, may consist of hundreds of distance constraints, each of them trying to maintain the given distance between neighbouring mass-points and, as a result, the given total rest-length of the whole instrument. When the tool is pushed or pulled at one end, it may take many iterations over the constraints before the solver converges and the other end responds.

3.1.2.4 Block solvers

Another approach to improve convergence rate is to mix global and local solvers. Blocks of constrained bodies are identified and, if needed, solved using exact methods in an iterative solver loop, hence the name – block solvers. A single block can contain, for example, a stack of rigid bodies, a whole deformable body or a set of mass-points linked in a chain-like structure, as described in the previous paragraph.

Free and open-source implementations of (block) iterative Gauss Seidel LCP solvers can be found under terms *Sequential Impulses* (Catto, 2005), *Quickstep* (Smith, 2007) or in Bullet Physics Library (Coumans, 2015). Other methods based on Conjugate Gradients (Silcowitz-Hansen et al., 2010) and Jacobi method (Frâncu, 2014) were also proposed.

3.1.2.5 Implementation aspects

An efficient and robust method for constraints solving is vital for inextensible rod simulation. At first, an attempt was made with a global LCP-like solver. Looking at

the Cosserat rod centreline as a system of constraints in a matrix form, it was recognised that they form a block sparse structure. By writing custom matrix-matrix and matrix-vector multiplication algorithms, short linear computation times were achieved. This was a significant improvement comparing to the naïve implementation on dense matrices. Additionally, re-arranging the order of constraints so that the collision constraints come after the distance constraints of corresponding mass-points, results in a symmetric, positive-definite, banded $\mathbf{JWJ}^T$ matrix (Figure 3.4). Solving banded matrices is again linear in time.

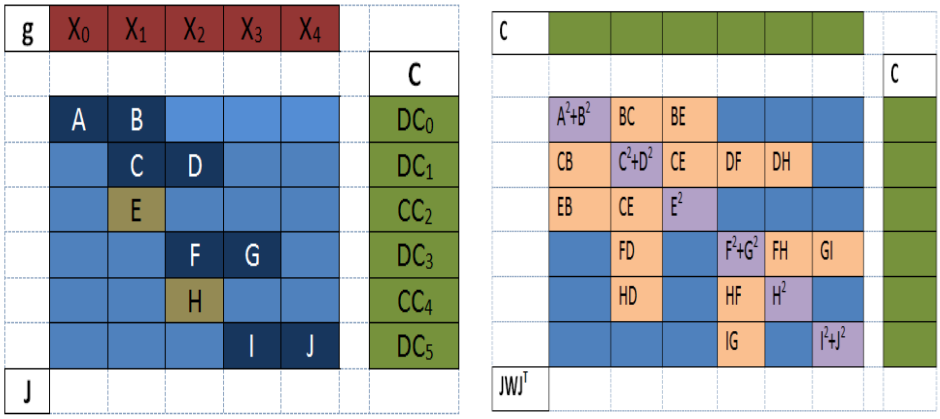


Figure 3.4: On the left: The rearranged J matrix. Navy blue are distance constrains (DC_i) and yellow are collision constraints (CC_j). On the right: The resulting symmetric banded JWJT matrix.

In this approach, the collision constraints were treated as regular equality constraints, using the sign of the Lagrange multiplier $\boldsymbol{\lambda}$ to keep or discard the constraint from the system. That is, after calculating the global $\boldsymbol{\lambda}$ vector holding the magnitudes of constraint forces, and knowing that positive $\boldsymbol{\lambda}$ generates forces that push the body away from the surface, whilst negative $\boldsymbol{\lambda}$ pulls its closer, it is possible to scan the $\mathbf{J}$ matrix to identify rows governing inequality constraints with respective negative $\boldsymbol{\lambda}$. If such rows do not exist, the final constraint forces $\mathbf{f}_c$ may be calculated. Otherwise, they can be simply removed from $\mathbf{J}$ and the computation of $\boldsymbol{\lambda}$ may be started again with the reduced set of constraints. A similar approach was used in (Kleppmann, 2007, Dequidt et al., 2007).

The second attempt was to use a block iterative Gauss-Seidel method with the distance constraints computed globally and collision and resting constraints processed locally working at the velocity level. The resulting tri-diagonal banded system of linear equations governing the distance constraints can be efficiently solved by a conventional linear solver as it contains only equality constraints. The LU factorization algorithms from the LAPACK library (www.netlib.org) were used. Next, the collision constraints including Coulombian friction are applied locally.

The block-iterative approach turned out to be faster, more stable, easier to implement and to extend. Due to the lack of stacked bodies and multiple contacts in the application domain, even a single iteration over the constraints was sufficient to achieve a visually plausible effect. Moreover, by slightly modifying the algorithm to be a non-block, i.e. fully iterative, by solving the distance constraints also locally, in a sequential manner, there is a possibility to massively parallelize the solver. All the constraints can be computed and applied on a per point basis as each distance constraint only needs access to the position and velocity of the two neighbouring mass-points. Of course, in this case there would be a need to increase the number of solver iterations to maintain the inextensibility of the rod. Despite this, a massively-parallel implementation can significantly improve the computational performance of rod simulation, which is presented in detail in Chapter 6.

3.1.3 Simulation loop

In summary, after a force is applied to the rod by a user manipulating it, for example with a haptic device, the bending energy V_b is differentiated and, if needed, the stretch energy V_s , and the parallel constraint penalty energy E_p are also differentiated with respect to the coordinates to obtain the stresses, i.e. restitution forces and torques which accelerate the centreline mass-points and material frames to equilibrium. Having the mass-points $\mathbf{r}_i$ loosely coupled by the parallel constraints with the quaternions $\mathbf{q}_j$ allows for their independent force integration. The mass-points are time integrated as if they were particles and the quaternions as if they were representing orientations of rigid bodies. After force integration, the

resulting velocities are constrained using the block-iterative or iterative solver, and integrated into new positions and orientations using a semi-implicit Euler scheme. Table 3.1 summarizes the simulation loop.

```
for i physics iterations (time-steps)
{
        detect collisions
        apply external forces and torques
        calculate Cosserat elastic forces and torques
        integrate forces and torques
        for j constraint solver iterations
        {
                apply distance constraints (globally or locally)
                apply collision response constraints
                if(selfCollisions)
                        apply self-collisions impulses
        }
        integrate linear and angular velocities
}
```

Table 3.1: The simulation loop

3.1.4 Collisions and self-collisions

A collision detection scheme was implemented running serially on the CPU or in parallel on the GPU. An axis-aligned bounding box (AABB) hierarchy guides the broad-phase collision detection stage, and a brute-force approach is used for the narrow-phase. Due to the hollow shape of the 3D models of human anatomy used in the application field (endovascular interventions, flexible endoscopy), the collision detection algorithm uses a bounding volume hierarchy, rather than a spatial partitioning such as BSP or OCT trees. This results in a well-balanced tree, which is more stable to traverse in terms of performance. The AABB tree is pre-computed at the initialization. During the broad-phase, the tree is searched recursively in $O(\log(n))$ time. Its size depends on the maximum depth of the tree, which is usually between 10 and 15 levels for complex anatomical models. In the narrow-phase, a brute-force collision check is performed against the triangle(s) in

the reported bounding boxes. Next, a collision response vector is calculated as a weighted average of all normal vectors of colliding triangles:

$$\mathrm{n_c} = \mathrm{normalize}\left(\sum_{i=0}^{c}\left(\mathrm{n}_i \cdot \frac{d_\mathrm{i}}{\sum d}\right)\right) \tag{18}$$

where n_i is the resulting normal vector of the colliding triangle $\boldsymbol{i}$, $\frac{\boldsymbol{d_i}}{\sum \boldsymbol{d}}$ – is the normalized distance to the $\mathbf{i^{th}}$ colliding triangle and $\boldsymbol{c}$ is the number of colliding triangles. The weight depends on the penetration depth for the given triangle. In the applications presented here, sphere-triangles collisions are used, but this scheme can support box-triangles and ray-triangles as well.

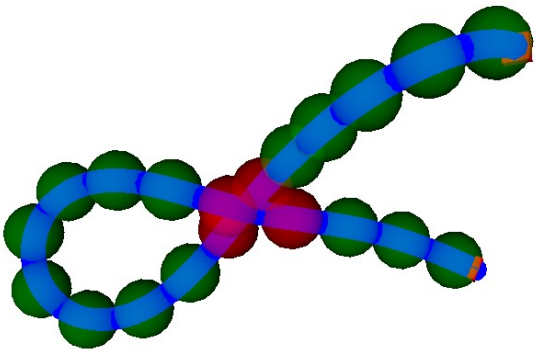


Figure 3.5: Self-collision detection: The mass-points of the centreline are coloured in blue. Larger green and red spheres are respectively non-colliding and colliding broad-phase bounding spheres.

In the case of self-collisions or collisions with another rod, a broad-phase stage based on bounding spheres wrapping a number of neighbouring mass-points (Figure 3.5) was developed. First, each bounding-sphere is checked against all others. Then, the colliding pairs are reported for a narrow-phase sphere-sphere check. If mass-points overlap, the collision response vector is stored to be used at the constraints solving stage.

3.2 RESULTS

3.2.1 Computational performance

Table 3.2 compares the performance of the proposed inextensible implementation on a HP x4600 workstation (Win7 x64, Intel Core2 Quad 2.66 GHz, 8GB RAM, NVidia GeForce GTX 560) to other models: the original extensible CoRdE (Spillmann and Teschner, 2007), inextensible modification by the same authors (Spillmann and Harders, 2010), the other inextensible elastic rod by (Bergou et al., 2008) and to the most recent Position Based Elastic Rod by (Umetani et al., 2014). Note that different test platforms were used and that the times for 100 mass-points in these papers are not given explicitly. In the case of (Spillmann and Harders, 2010), the times were derived from the coil embolization example where authors state constituent times of a simulation of 40 Cosserat mass-points. Adding these and linearly extrapolating from 2.26ms per 40 points gives an approximated time of 5.65ms for 100 points. In the case of (Bergou et al., 2008), the times were linearly extrapolated from tests 7 (0.34ms / 75 points) and 8 (0.42ms / 67 points). Umetani et al. (Umetani et al., 2014) gives the computation times just for one test (1.06ms / 30 points).

Model	PC CPU	Time stated in paper (ms/pts)	Time (ms/100 pts)	Speed-up
Our inextensible model	Core2 2.66 GHz	0.147/ 100	0.147	x1.00
Original CoRdE (Spillmann and Teschner, 2007)	Xeon 3.80 GHz	0.131 / 100	0.131	x0.89
Inext. CoRdE (Spillmann and Harders, 2010)	Core2 3.00 GHz	2.26 / 40	5.65*	x38.4*
Discrete Elastic Rods (Bergou et al., 2008)	Core2 2.66 GHz	0.34/75 - 0.42/67	0.45 - 0.67*	x3.06-x4.56*
Position Based Elastic Rods (Umetani et al., 2014)	N/A	1.06 / 30	3.53*	x24.0*

*Table 3.2: Comparison of computational times with other models. *Approximated times.*

The modified model proposed in this chapter was only 0.016ms slower than the original, stretchable CoRdE presented by (Spillmann and Teschner, 2007). However, assuming that the approximations are correct and considering only the inextensible rods, the modified CoRdE presented here was roughly x4, x38, x24 times faster than the approaches by (Bergou et al., 2008), (Spillmann and Harders, 2010) and (Umetani et al., 2014), respectively. Still, these models have other advantages,

primarily, the elimination of the penalty method in parallel constraints, improved stability or improved contact handling.

3.2.2 Inextensibility

Figure 3.6 presents the change in length and resulting rod stretch under different type and number of distance constraints iterations: a single iteration of global constraints (1G), multiple (1-50L) iterations of local constraints, as well as stretch penalty forces applied as in the original CoRdE model (1I, for internal). The test environment consisted of rods of different lengths (128, 256, 512 and 1024 elements) locked at one end and released like a pendulum, swaying for a few seconds in free space. The time-step was set to 5ms, 2.5ms and 1ms, the distance constraint mixing factor (k_d) to 1.0 and the Baumgarte stabilization term (k_s) to 0.1. In the case of penalty stretch forces, E_s was set to the highest stable value that could be achieved for the given time-step. No collision detection or self-collisions were involved.

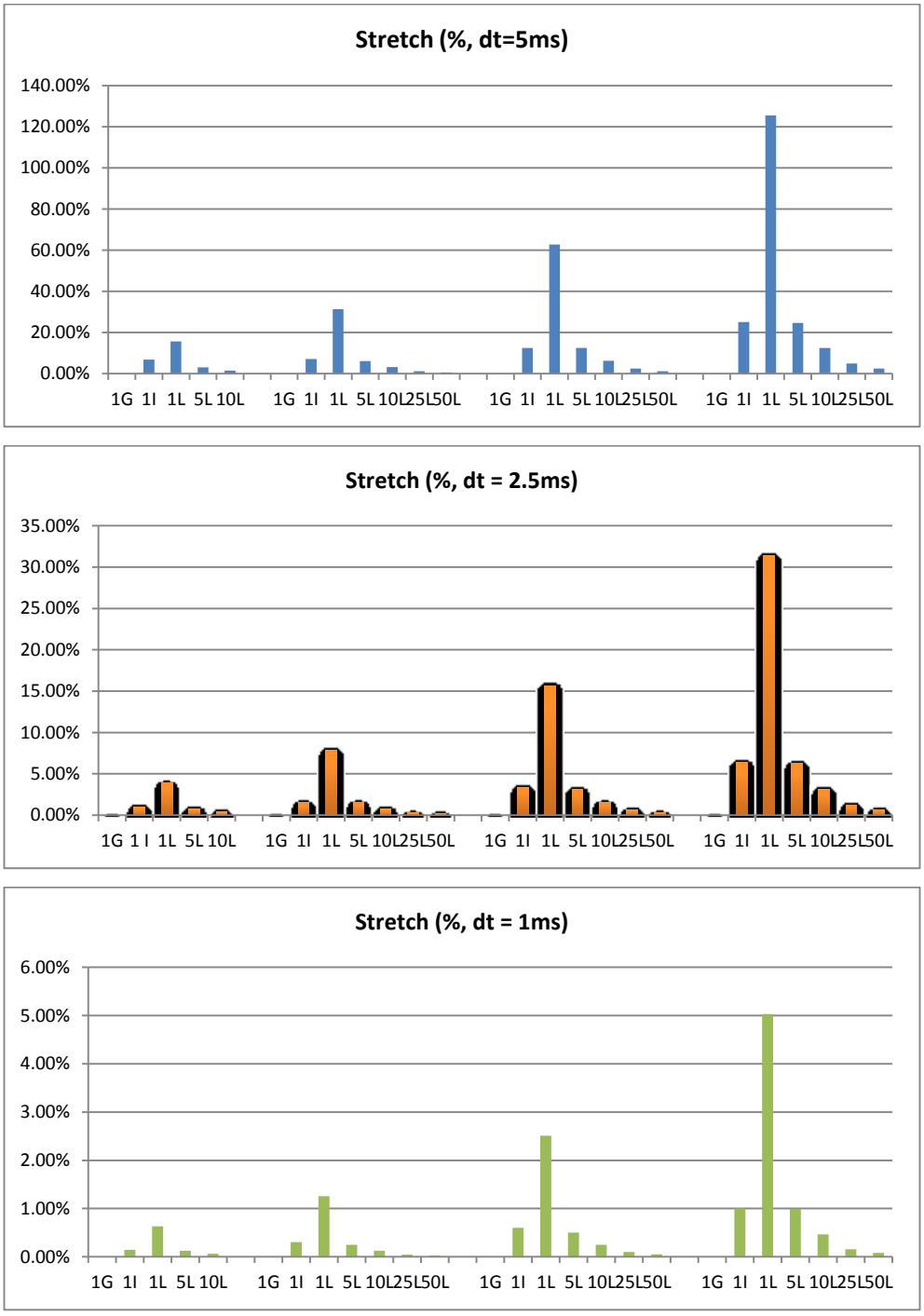


Figure 3.6: Percentage of stretch of the rod under different type and number of distance constraints iterations for rods consisting of 127, 256, 512 and 1024 mass-points.

From this experiment, it can be concluded that a single iteration of a global distance constraint managed to completely eliminate the stretch for all rod lengths and for

all time-steps. These two factors had a crucial impact on the accuracy of the other constraints. The stretch grew linearly in respect to the rod length, i.e. number of mass-points, and quadratically in respect to the time-step. The local iterative method requires roughly 5 iterations to reduce the stretch to the level of the original CoRdE using penalty forces for stretch calculations.

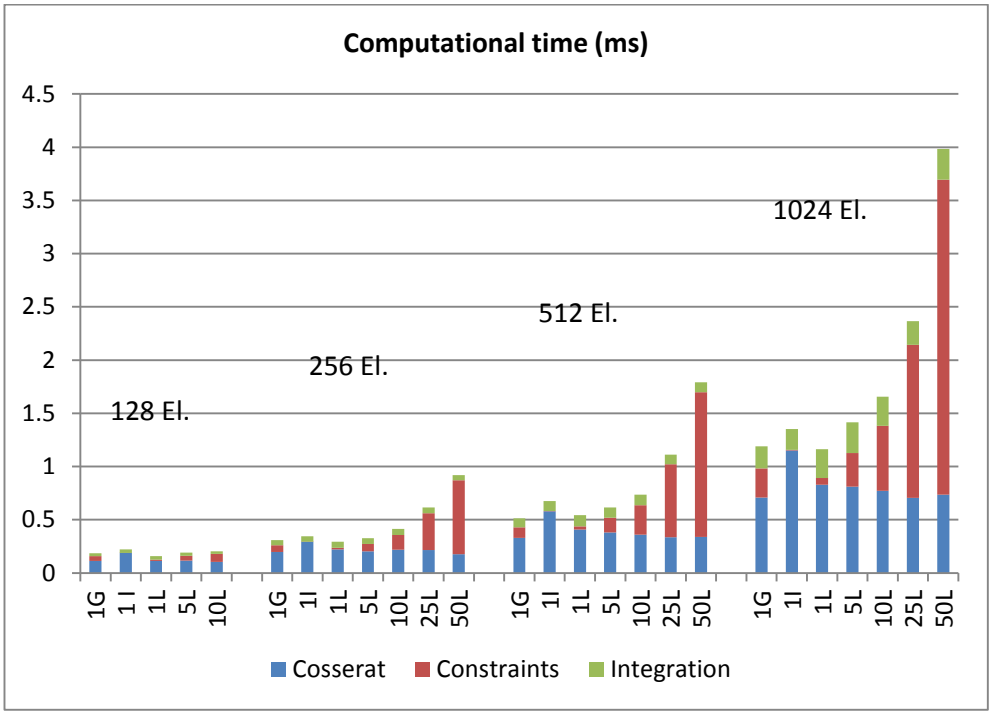


Figure 3.7: Computational times of different type and number of distance constraints iterations

Figure 3.7 shows that increasing the number of local iterations comes at an additional computational cost growing linearly depending on the number of Cosserat elements. A single global iteration is only marginally slower than a single iterative.

Section 5.4.2 in Chapter 5 provides an insight into the compressibility of the rod.

3.2.3 Real-time interactions

In these qualitative tests (Figure 3.8 and Figure 3.9), two Phantom Omni haptic devices (www.geomagic.com) were used to control the loose ends of the rod. The proxy objects, linked to the rod's ends by damped springs, were steered directly by

the devices. A model of random vascular anatomy was placed in the scene allowing for collisions. The rod also self-collided.

The behaviour of the virtual rod was visually plausible. The rod was stable and responsive. The stretching was unnoticeable and both collisions and self-collisions held well even for relatively high forces, allowing for wrapping and tying complex tight knots on the vascular model as shown in Figure 3.8 on the left. Applying unnatural, excessive forces was causing instabilities (shaking) of the rod and eventually led to an explosion of the model.

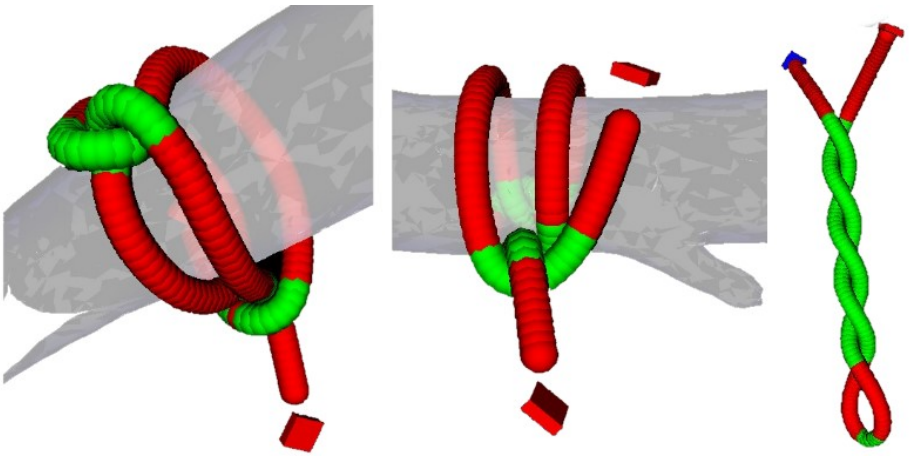


Figure 3.8: On the left and in the middle: The Cosserat Rod wrapped around a polygonal model. On the right: A plectoneme formed by multiple twisting of two rod ends in opposite directions.

By locking the quaternions at the either end of the rod, it was possible to twist the ends of the rod in opposite directions. After a few twists, the rod started to loop and started to form a plectoneme as shown in Figure 3.8 (right) demonstrating the aforementioned coupling between bend and twist deformations.

Figure 3.9 shows two stages of tying a double Fisherman's knot similarly to the test found in (Spillmann and Teschner, 2008). Two Cosserat rods consisting of 256 mass-points each were attached to different points in space at only one end. The Omnis controlled the loose ends allowing for manually tying the knot. In the tied knot, 300 collisions and self-collisions occurred on average. The tied knot configuration took 2.67ms to compute per single iteration. This is below haptic interactive rates (<1ms)

and shows that even some, as it would appear, fairly simple simulations can be problematic in real-time. This problem is addressed in Chapter 6.

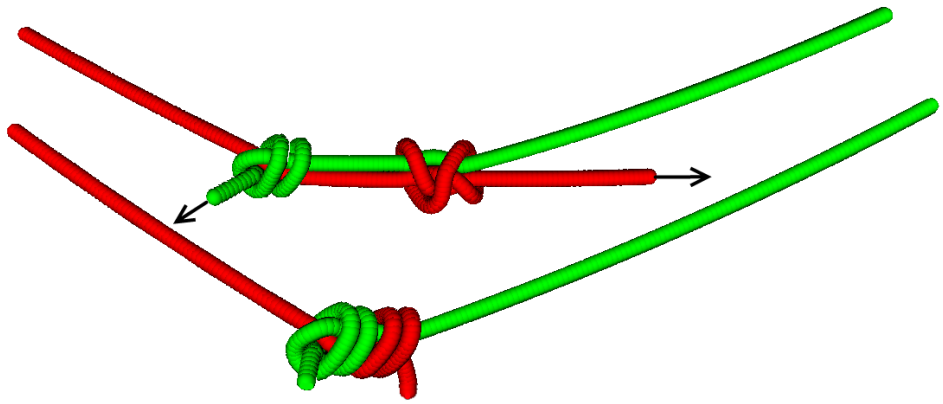


Figure 3.9: Tying a double Fisherman's knot.

3.3 SUMMARY

This chapter presented the continuous and discrete mathematical formulations of the CoRdE model. The CoRdE model is based on the Cosserat theory of elasticity. Its computer implementation uses explicit centreline representation to model material stretch and material frames adapted to the centreline by penalty forces, to recreate the bending and twisting phenomenon. Following the original CoRdE formulation, the bending V_b and stretch V_s energies were derived which, differentiated with respect to coordinates, yield the restitution forces and torques. These accelerate the centreline mass-points and material frames to equilibrium.

The proposed modification to the CoRdE model efficiently maintains rod inextensibility by using a block iterative constraints solver. The parallel constraints, which align the material frames to the centreline, are applied using the penalty method yielding an additional energy term, the same way as in CoRdE. However, the penalty method governing the stretch in centreline of the original model was replaced by a new constraints formulation. The distance constraints arranged in a tri-diagonal system of linear equations are solved exactly in a linear time. Next, the collision constraints including Coulombian friction, which prevent the rod from penetrating other objects, are applied locally in an iterative manner.

In a series of quantitative and qualitative experiments, it was shown that the proposed virtual rod model achieves real-time performance at haptic interactive rates (>0.5-1kHz), even for longer rods (i.e. >500 mass-points), whilst being non-stretchable, stable and visually plausible. In the author's opinion, this approach offers a good compromise between computational efficiency (Spillmann and Teschner, 2007) and simulation accuracy (Spillmann and Harders, 2010). Rod inextensibility and incompressibility was efficiently enforced but, since in the chosen applications there is no need for stiff rods, and considering simplicity, performance and reasonable accuracy, the penalty method was left intact. A possible disadvantage of this approach is the need for an additional numerical spring constant parameter K_p influencing the behaviour of the rod.

In summary, the modified CoRdE implementation enables real-time simulation at haptic interactive rates, an immediate response to user manipulations at the proximal end, efficient twisting and collision handling, and an easy parameterization of the mechanical properties of the rod. Hence, the model established solid foundations for its application to the simulation of flexible surgical instruments. This will be the subject of the following two chapters.

Chapter 4

NOViSE - A VR SIMULATOR FOR NOTES SURGERY

Natural Orifice Transluminal Endoscopic Surgery (NOTES) is a novel technique in minimally invasive surgery (MIS), whereby a flexible endoscope is inserted via a natural orifice to gain access to the abdominal cavity, leaving no external scars. This innovative use of a flexible endoscope creates many new challenges and, due to the rudimentary properties of the endoscope, is associated with a steep learning curve for clinicians. This chapter introduces a prototype virtual reality simulator for NOTES. The simulator supports a complete trans-gastric hybrid cholecystectomy operation. The behaviour of the virtual flexible endoscope is modelled based on an established theoretical framework – the Cosserat rod – presented in Chapter 3. VR simulation of NOTES procedures can contribute to training in this new surgical technique without putting patients at risk, raising ethical issues or requiring expensive animal or cadaver facilities whose use and acceptability may be limited.

The following give a brief introduction to NOTES (section 4.1), related work (section 4.2) and present the Natural Orifice Virtual Surgery (NOViSE) simulator (section 4.3), including its overall design (4.3.1), haptic interface (4.3.2), virtual flexible endoscope model (4.3.3), camera, light source and actuators (4.3.4), tool-tissue interactions (4.3.5), as well as multi-threaded implementation (4.3.6). This is followed by details of its application to a hybrid trans-gastric cholecystectomy procedure (4.4.1) and extensive face, content and construct validation results (4.4.3).

4.1 NATURAL ORIFICE TRANSLUMINAL ENDOSCOPIC SURGERY

Over the last 30 years, laparoscopic surgery has become the standard approach for many operative procedures. In order to push minimally invasive techniques further along the spectrum towards truly non-invasive surgery, surgeons have started using flexible endoscopy in procedures traditionally reserved for rigid instruments. By inserting a flexible endoscope via a natural orifice such as the oesophagus, vagina, or anus (Figure 4.1), and then navigating the endoscope through an internal incision in the relevant organ, surgeons can gain access to the abdominal cavities and are able to, for example, remove the gallbladder (cholecystectomy) leaving no external scars (incision/scar-less procedure). This emerging technique is known as Natural Orifice Transluminal Endoscopic Surgery (NOTES). Since it eliminates external postoperative wounds, it is argued that NOTES may further reduce operation trauma, recovery time, clinical costs and improve overall cosmetic results, thereby pushing the boundaries of minimally invasive surgery as we know it (Spivak and Hunter, 1997, MacFadyen and Cuschieri, 2005, Richards and Rattner, 2005, Ponsky, 2006). As with any new, potentially disruptive surgical technique, the benefits of NOTES are still to be fully realised, and there remains considerable dissent as to its true benefits and risks (Rattner et al., 2011, Moris et al., 2012).

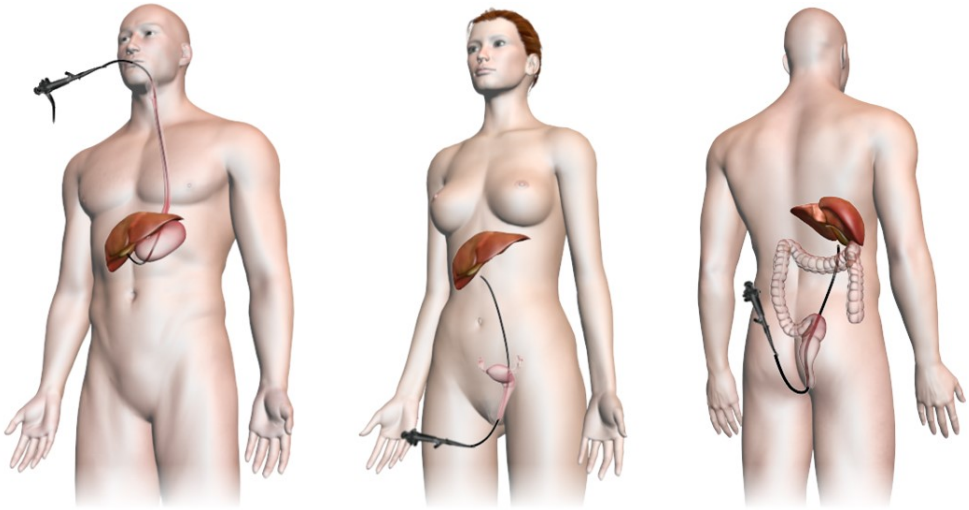


Figure 4.1: Trans-gastric (left), trans-vaginal (middle) and trans-rectal (right) NOTES approaches

The first human NOTES operation was carried out in April 2007 at the University Hospital of Strasbourg by Professor Jacques Marescaux and his team (Operation "Anubis"), where they successfully performed a hybrid trans-vaginal cholecystectomy on a female patient with symptomatic gallstones. Although NOTES has been gaining popularity among surgeons and patients around the globe since then, it is still considered an experimental technique associated with significant challenges that must be addressed before widespread clinical adoption is made possible, particularly as the procedures themselves require great technical expertise. The Natural Orifice Surgery Consortium for Assessment and Research (NOSCAR) established a list of potential barriers which need to be surpassed before NOTES can be incorporated in routine practice (Rattner, 2006). One of the key issues identified by specialists was the lack of efficient training programs available for clinicians and the extremely steep learning curve of NOTES procedures. NOSCAR also pointed out that use of conventional gastro- and colonoscopes in NOTES procedures is far from being optimal. Recently, a couple of prototype devices aimed specifically at NOTES have been announced (Figure 4.2). However, to the author's knowledge, none of them have reached the commercialization stage.

Figure 4.2: Prototypes of NOTES endoscopes. On the left, Cobra by USGI medical. In the middle, a concept device by Olympus Medical Systems. On the right, Anubis by Karl Storz.

The novel use of a flexible endoscope in NOTES procedures differs substantially from both conventional endoscopy and laparoscopy. In addition to entering the abdomen through a natural orifice, the NOTES technique requires the surgeon to operate the endoscope and any associated instrumentation through a single access

point, rather than the three or four ports common to a laparoscopic procedure. Although many NOTES procedures are currently being performed in a hybrid fashion (i.e. with some trans-abdominal assistance), there is a significant loss of retraction and the in-line instrument approach through the instrument ports in the endoscope handpiece is unfamiliar to most surgeons. Another significant difference is the lack of a gastrointestinal lumen to support the endoscope. The distal end of the endoscope is manipulated in the open abdominal cavity using the incision (viscerotomy) site, internal organs and gravitational force to navigate and position the instrument. The middle section of the endoscope shaft can imperceptibly roll and loop inside the abdomen. As a result of these differences in endoscope behaviour, the approach to regions of interest can also be very different in comparison to traditional endoscopic techniques. Taking all of the above into consideration, it is clear that performing NOTES procedures requires a new set of skills. Learning and practicing these new skills demands a new set of training tasks supported by suitable simulator models.

The most recent survey on education and training in NOTES (Moghul et al., 2013) reviews 11 non-animal studies, 8 animal studies and 6 educational programs for NOTES. Several of them demonstrate construct validity (the ability to differentiate between expert and novice operators). Most notable is the "ELITE" simulator - an ex-vivo, full-scale replica of a female adult with various transluminal access points (Gillen et al., 2009, Gillen et al., 2011). The survey also states that minimal work has been carried out in the field of Virtual Reality (VR).

4.2 RELATED WORK

A recent needs analysis for a NOTES VR simulator shows that there is indeed interest in VR technology for NOTES (Sankaranarayanan et al., 2013). Whilst there are well-established, validated commercial VR simulators available for flexible endoscopy procedures (*GI Mentor* - *www.simbionix.com*, *EndoVR* - *www.caehealthcare.com*) such as Lower/Upper GI, Endoscopic Retrograde Cholangiopancreatography (ERCP), Endoscopic Ultrasound (EUS) or Flexible

Sigmoidoscopy (FS), simulating NOTES procedures requires more advanced dynamic modelling of the virtual endoscope so that it may operate in open abdominal cavities, and interact with surrounding anatomy in a different manner.

In (Ahn et al., 2014), Ahn and colleagues report on-going work on their *Virtual Transluminal Endoscopic Surgery Trainer - VTEST*. Their simulator aims to recreate a hybrid NOTES procedure using a rigid scope and a trans-vaginal approach. In (Dargar et al., 2014), the same group describes their work on a prototype haptic device for flexible endoscopy, but this is yet to be integrated into their *VTEST* system and no results of its performance are presented. Therefore, to the best of the author's knowledge, there are currently no force-feedback enabled, either commercial or experimental, VR simulators for NOTES procedures supporting a flexible endoscope.

4.3 METHODS

4.3.1 Simulator overview

The simulator set-up (Figure 4.3) consists of a real-time software simulation and a physical, force feedback human-computer-interface (haptic device). The software is written in Java, with performance critical sections in C/C++, and can efficiently run, i.e. exceeding haptic interactive rates, on a modern mid-range PC or laptop with Windows, MacOS or Linux operating systems. The simulator display is divided into two parts. On the right, the user can see the endoscopic camera view. On the left there is an external, optional "aerial view" which can be turned on/off and freely manipulated.

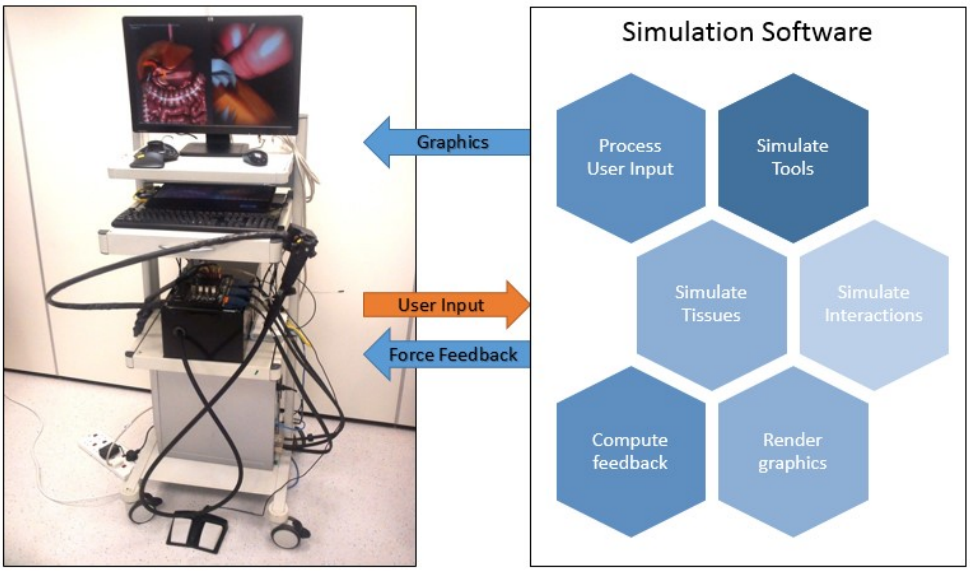


Figure 4.3: On the left: the complete NOViSE set-up. On the right: the overview of the simulation software sub-systems.

4.3.2 Haptic device

The haptic device was designed (Figure 4.4) and built for the purpose of this project by Dr Alastair Barrow. It comprises an enclosed black box of dimensions approximately 55x26x18cm, into which passes a hose (1.5m long, 15mm diameter, Figure 4.5 left). The hose can be pushed or pulled through the opening (total travel 22cm) and rotated freely. Inside the enclosure, the end of the hose is directly

coupled to a 15:1 planetary gearbox and a servo motor delivering a combined total torque of +/- 2.55Nm. This motor is mounted on a low friction linear rail driven by an identical motor connected via a tensioned toothed drive belt and a 24mm pulley resulting in +/- 14N linear force output. Both linear (14N) and rotational (2.55Nm) force feedback significantly exceed the requirements for endoscopic NOTES procedures. These were measured by (Dargar et al., 2014) and peak at 4.77N for linear and only 0.03Nm for rotational feedback.

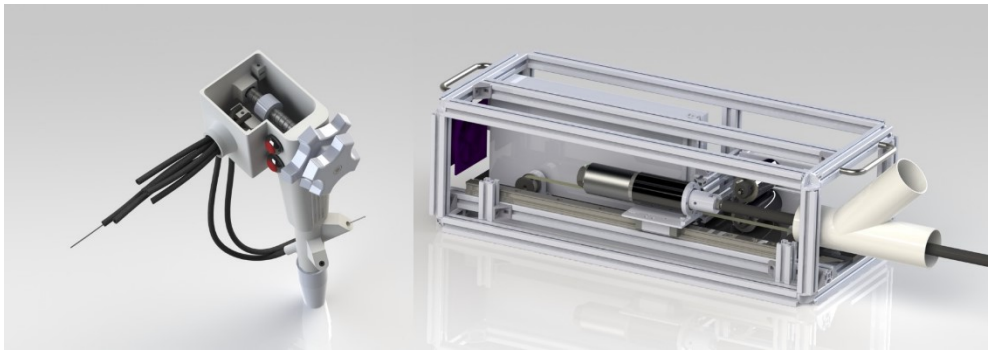


Figure 4.4: The CAD rendering of the haptic device designed and built by Dr Alastair Barrow.

At the proximal end of the hose, a 3D printed plastic replica of a standard endoscopic handpiece is attached (Figure 4.5 right). It consists of two force-feedback enabled thumb wheels, two optically tracked thin wires representing the endoscopic tool wires and two push buttons. Additionally, a double foot pedal is placed on the floor and can be used to activate endoscopic instruments, e.g. diathermy.

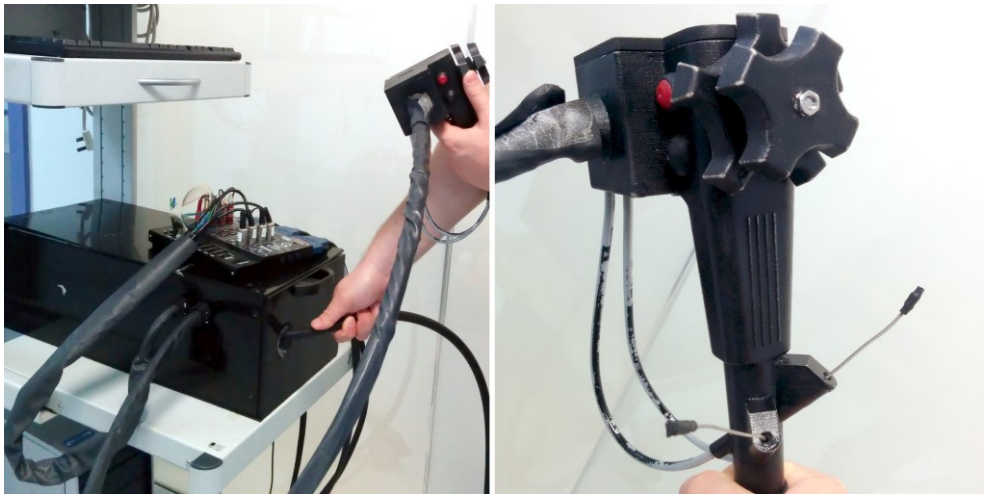


Figure 4.5: Left: The haptic device connected to the data acquisition device. Right: A close-up to the hand piece.

The haptic device is connected to a *QPID* (*www.quanser.com*) data acquisition board, which communicates via a PCIe interface with a desktop PC (server). The server reads the state of the haptic device sensors, processes it and sends it to the simulation software (client) using the UDP protocol via a local gigabit Ethernet network. The simulator responds to the server with a packet commanding the motors responsible for the force-feedback. This communication runs at a rate of 1kHz.

4.3.3 Virtual flexible endoscope model

The flexible endoscope model is based on the inextensible Cosserat Rod (Spillmann and Teschner, 2007) introduced in Chapter 3. The physical properties of the virtual endoscope, such as mass, diameter, resistance to bending and twisting were derived manually, under the supervision of an expert clinician, to match the behaviour of an existing model. The operator controls the virtual endoscope using the haptic device. The shaft of the virtual endoscope can be pushed, pulled and rotated through manipulating the haptic device. The tip of the virtual endoscope is steerable in two directions, with it's $\hat{u}_1$ and $\hat{u}_2$ intrinsic bend parameters controlled by two thumb-wheels on the handpiece of the haptic device.

The virtual endoscope is equipped with a light source, a camera and two working ports through which different instruments (actuators, Figure 4.6) may be inserted. Currently, users can choose from four types of virtual actuators: grasper, clipper, scissors and diathermy tool. Their insertion / removal is controlled by two physical wires inserted in the two ports of the handpiece. When the wires are fully retracted, the actuators may be swapped for different ones by pressing one of the buttons on the handpiece. The virtual actuators are activated by pressing on a foot pedal.

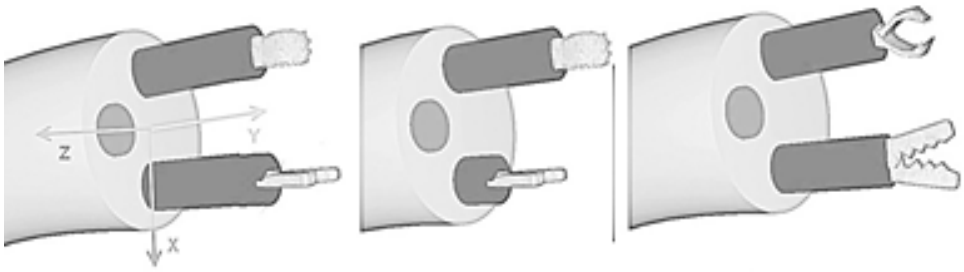


Figure 4.6: Endoscope tip, camera and two actuators: a grasper and cutter.

Similarly to the traditional gastro- and colonoscopes used currently in NOTES procedures, the haptic interface allows only for pro- and detruding of the actuators. However, as the actuators are also simulated using the Cosserat Rod, their articulation can be increased by controlling their bend and/or rotation. This allows for recreating a range of prototype endoscopes (Figure 4.2) designed specifically for NOTES. Additionally, by placing the endoscopic camera at the tip of a separate actuator, it is possible to achieve a triangulation – a “natural” configuration of camera and instruments, shown in Figure 4.7, which is preferred by the operators.

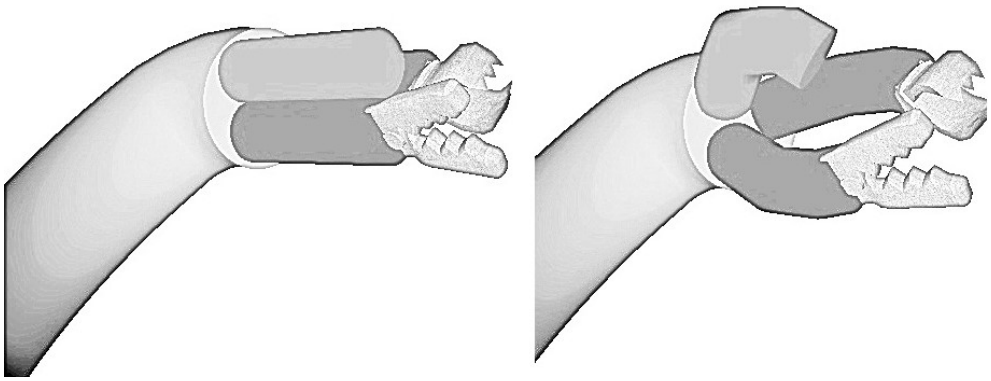


Figure 4.7: The straightened (left) and triangulated (right) actuators

This supplementary control is, however, not available via the current haptic interface. As it requires an additional analog pad for fine actuator steering and none of the NOTES specific endoscope is currently available, it was decided to include only traditional endoscopes in the validation process in Section 4.4.3.

4.3.4 Camera, light source and actuators

The simulator virtual environment and visualisation is implemented using *JME3* - a Java OpenGL 3D graphics engine. It is divided into two parts (Figure 4.8). On the left side of the screen, there is an optional "aerial view" which can be freely manipulated. On the right, the user can see a picture from a camera fixed at the tip of the virtual endoscope. Next to this camera a bright spot-light source is located. To recreate the relevant abdominal organs, detailed off-the-shelf commercial 3D anatomical models (*www.3dscience.com)* were used. They include the oesophagus, liver, stomach, gallbladder, pancreas, spleen, duodenum and small and large intestines. The organs are illuminated using a per-pixel lighting shader implemented in GLSL and textured using diffuse textures and normal maps. The spot-light casts soft shadows using a Parallel Split Shadow Mapping (PSSM) technique to further increase the depth perception. A radial blur and bloom effect (blurring of bright areas of a scene) are applied during the post-processing stage to smooth out the final picture and mimic the imperfect quality of actual optics.

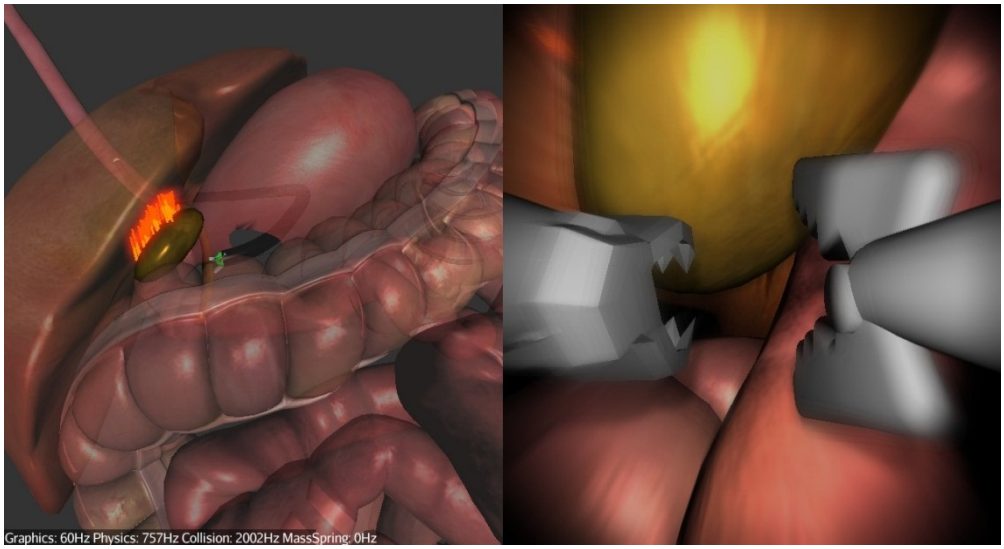


Figure 4.8: Simulator display. Left: auxiliary external view. Right - endoscopic camera view with visible grasper and cutter.

4.3.5 Tool-Tissue Interactions

For performance, only the target operative organs are modelled as deformable objects, with all other organs treated as non-deformable, but the endoscope and its actuators still respond to collisions with them. In the case of a cholecystectomy, only the gallbladder is considered to be deformable and modelled with a mass-spring model (Meier et al., 2005, Nealen et al., 2006). TetGen (*www.tetgen.berlios.de*) was used during the pre-processing phase to generate a tetrahedral mesh from the polygonal model of the gallbladder (Figure 4.9), consisting of 1194 mass-points, 3615 tetrahedrons and 5914 connecting springs.

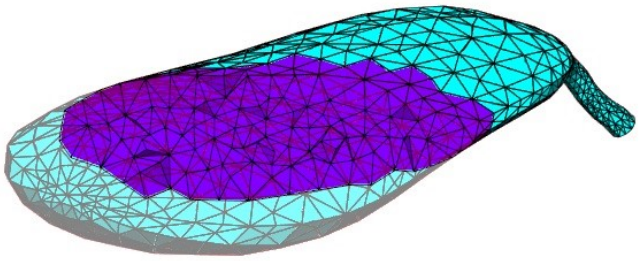


Figure 4.9: A cut through a tetrahedral mesh of the gallbladder generated using TetGen.

Its physical properties, such as spring stiffness and dampening, were tuned manually, following the judgement of NOTES experts, to approximate the behaviour of the real anatomy, whilst achieving real-time performance. The gallbladder deforms as it interacts with the endoscope and the actuators. Its body can be probed, grasped, and its cystic artery and duct can be clipped and cut. Figure 4.10 presents the behaviour of the gallbladder during stretching. Additionally, it can be retracted using a rigid laparoscope in a conventional way (hybrid NOTES procedure). The connective tissue between the gallbladder and the liver can be dissected using the diathermy tool or scissors.

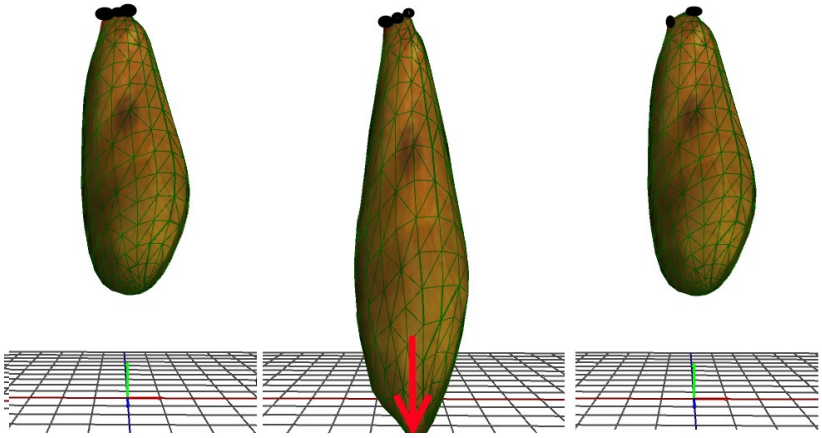


Figure 4.10: Gallbladder deformations: hanging neutral (left), pulling down hard (middle) and going back to resting shape (right)

Collision detection between the endoscope or the actuators and surrounding tissues is performed using the collision scheme presented in Chapter 3. An axis-aligned bounding box (AABB) hierarchy guides the broad-phase collision detection stage, and a brute-force approach is utilised for the narrow-phase collision detection. The collision response uses weighted average of normal vectors of all colliding triangles. The resulting direction vector is then used by the constraints mechanism to prevent the mass-points from moving along this direction. For performance reasons, only tissues which are in the region of interest of the operator, i.e. reachable by the endoscope (Figure 4.11), are contained in the BVH.

.

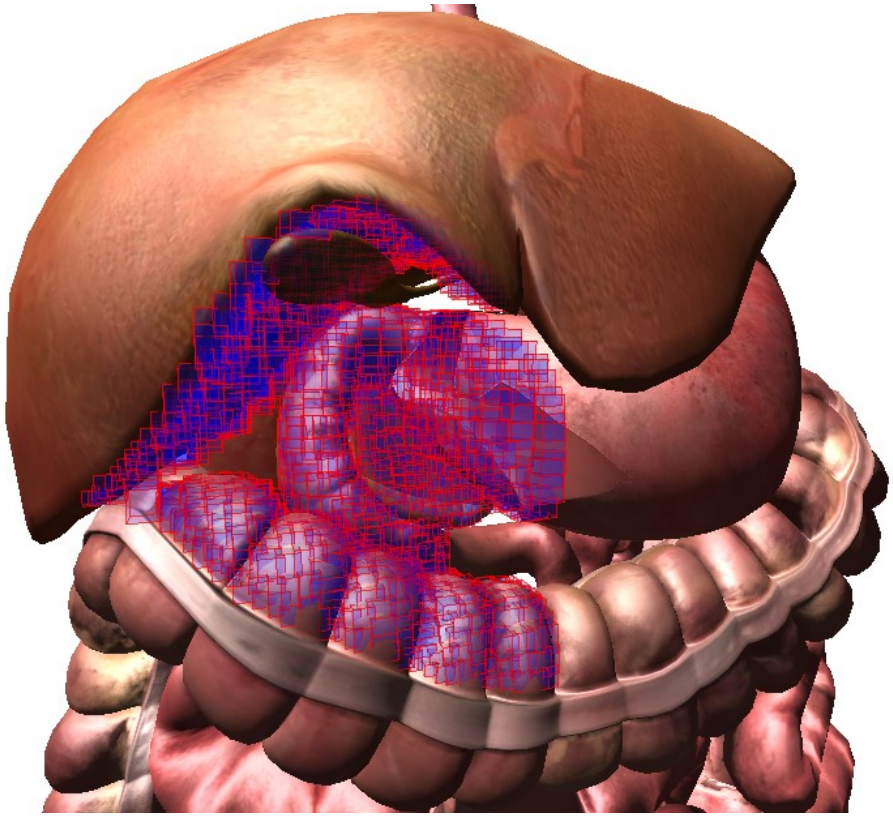


Figure 4.11: The abdomen tissues reachable by the endoscope wrapped in an AABB BVH tree.

The collisions between the gallbladder and the surrounding anatomy or gallbladder self-collisions are implemented by assigning each mass-point of the dynamic deformable body a small radius and performing a sphere vs. triangle check against the static or dynamic surrounding mesh or itself. This is computationally intensive and, depending on the platform, can be omitted if needed.

4.3.6 Multi-threaded implementation

In order to harness the power of modern CPUs, the software exploits multiple cores. There are two common approaches to parallelizing software: data parallelism and task parallelism. Data parallelism distributes the data, in this case the mass-points and quaternions, across different CPU cores. For example, given a quad-core CPU and 100 mass-points, each core would fully process a section of 25 neighbouring mass-points. This processing for each mass-point in a section includes

calculation of Cosserat internal forces, collision detection, calculation of constraints, integration and scene-graph transformations for 3D rendering. Task parallelism distributes the tasks across different CPU cores. In the previous example, the first core would be dedicated to calculations of only internal forces and constraints of all mass-points, the second core only to collision detection, the third core to scene-graph transformations, and the fourth core to handling mass-spring models of organs. If needed, the collision detection task, due to its locality, can be distributed further to a larger number of threads. Experiments showed that, in this case, task parallelism results in better performance compared to data parallelism. This is caused by a smaller synchronization overhead, fewer CPU context switches and higher cache utilization (recent Intel CPUs have L1 caches per-core and L2 cache that is shared among the cores).

Different tasks can run at different rates relatively to the physics task. The collision detection task runs in parallel, one to one, a step behind, but the deformable body task runs at 1/3 rate and haptics at 1/4 of the instrument physics. A form of double buffering is used to reduce the synchronization overhead and prevent concurrency issues such as visual flickering or physics instabilities.

4.4 RESULTS

The application of the above methods to simulate a hybrid transgastric cholecystectomy procedure, including a series of proposed performance metrics and computational performance of the NOViSE simulator are now presented, followed by the results of a face, content and construct validity study.

4.4.1 Hybrid Transgastric Cholecystectomy

Hybrid trans-gastric cholecystectomy was chosen since laparoscopic cholecystectomy (gallbladder removal) is one of the most prevalent surgical interventions, it was also amongst the first NOTES procedures and it is currently how it is performed in NOTES clinical practice. It is a hybrid operation as laparoscopic assistance for visualization and retraction is coupled with a flexible endoscope.

The simulation starts with the endoscope partially inserted into the oesophagus. It is divided into three main tasks: navigation via the stomach to the abdomen (Figure 4.12), clipping and cutting of the Calot's triangle (Figure 4.13) and gallbladder dissection using the diathermy tool and the grasper (Figure 4.14). The operator is guided by glowing markers indicating an optimal path, an incision (viscerotomy) site, clipping points/angles and connective tissue.

During the first task (Figure 4.12), the operator needs to find the viscerotomy located on a side of the stomach and navigate the scope through it into the abdominal cavity. S/he is not required to pierce the stomach as the incision is already present and represented by a glowing red ring. However, navigating through the ring is not trivial as it requires a combination of bimanual motions of the endoscope shaft and handpiece controls.

Figure 4.12: Task 1 - Navigation from the stomach into the abdomen via the incision (viscerotomy) site (glowing red ring)

After entering the abdomen, the operator can proceed to the second task - clipping and cutting of the cystic artery and duct (Figure 4.13). The operator needs to locate an anatomical region called Calot's triangle and start by clipping the cystic artery first. The optimal clipping point is indicated by a blue marker where the operator must insert a clipping tool, positioning its jaws as close to the blue marker on the artery as possible, while maintaining a right angle between the jaws and the artery. Once the required location is reached, the blue marker will glow, with its brightness dependant on how close the angle between the artery and the jaws is to the ideal (90 degrees). After positioning the jaws, the operator can deploy a clip by pressing the button on the handpiece or, if preferred, on the foot pedal. Next, s/he needs to place another clip on the artery and cut between the clips using scissors. This clip and cut process is repeated on the cystic duct. The key to completing this task efficiently is to correctly navigate and position the tool right from the start so that all the clipping and cutting can be done without having to manipulate the endoscope. This way, all the points of interest should be within reach by just adjusting the tip of the scope and inserting / removing the actuators.

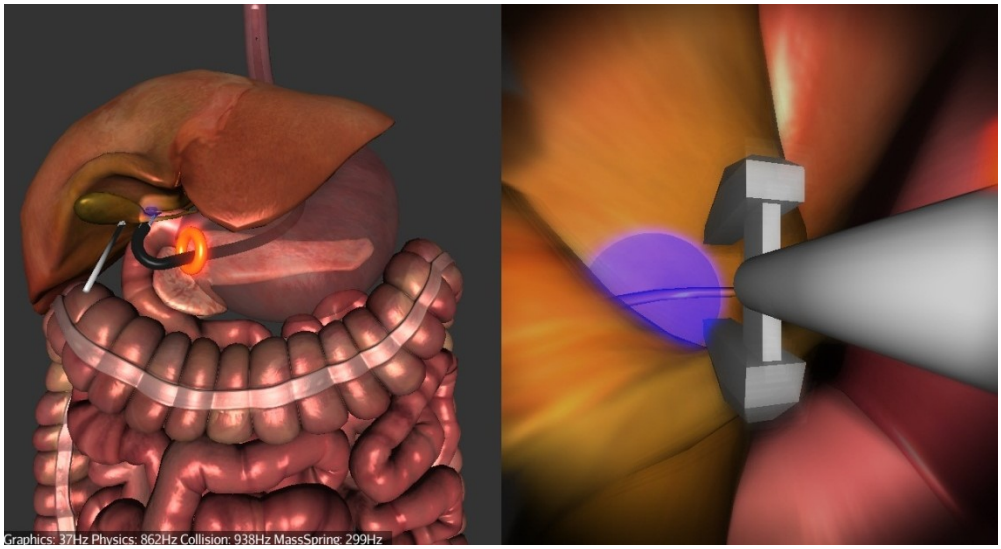


Figure 4.13: Task 2 - Clipping the cystic artery

Having clipped and cut the cystic duct and the artery, the operator can progress to dissect the connective tissue between the gallbladder and the liver bed using the diathermy tool (Figure 4.14). The connective tissue is represented by red glowing line segments that are burnt by the operator activating the diathermy tool close to them. Activation of the diathermy needs to be precise and accurate in order to burn as little of other non-target tissues as possible.

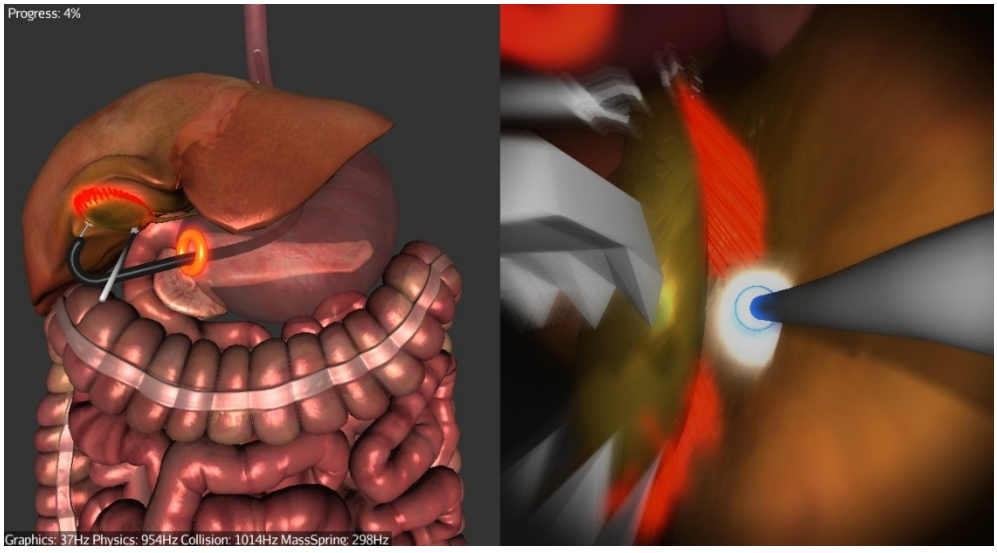


Figure 4.14: Task 3 - Gallbladder dissection using diathermy. The red line segments represent the connective tissue. In the back, an auxiliary laparoscope retracting the gallbladder is visible.

As mentioned above, the simulated procedure is hybrid, which means that there is still one laparoscopic instrument deployed in the conventional way which is used to retract the gallbladder in order to get a better exposure of the connective tissue. In order to prevent bias, this retraction is controlled using the keyboard by the assistant following a direct voice command from the operator. After removing the connective tissue, the operator can use the actuator grasper to hold the gallbladder and pull it out through the stomach (Figure 4.15). At this point the procedure is completed. The operator is not required to close the viscerotomy site.

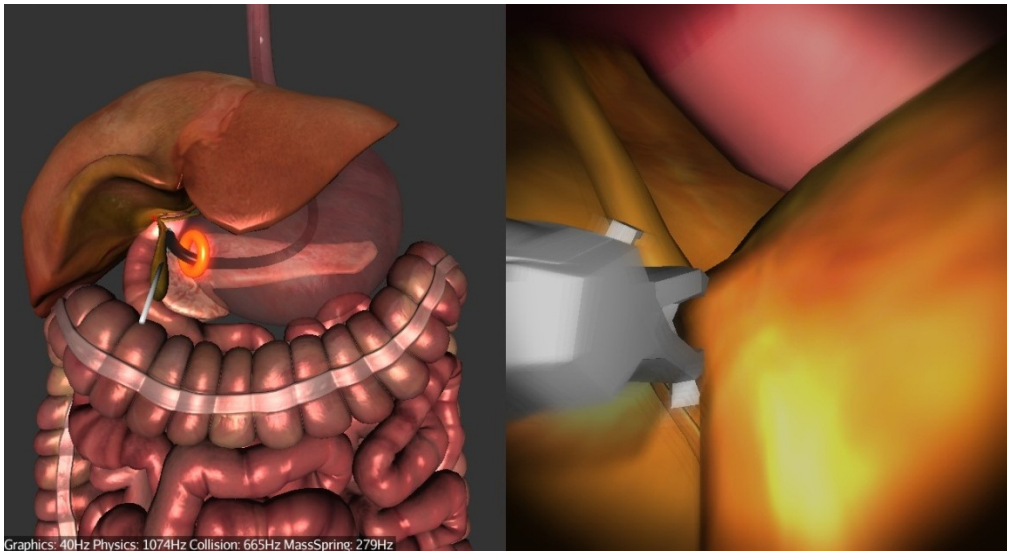


Figure 4.15: Completing the procedure by pulling out the gallbladder using the grasper.

In between certain tasks, the screen occasionally fades out and the simulation is paused. This is because the length of the hose (flexible endoscope) is less than that of a real endoscope. When this happens, the operator is asked to adjust the insertion of the hose so that s/he will have enough insertion/retraction available to complete each task without reaching the limit of the haptic device.

4.4.2 Metrics

The software computes a series of metrics related to particular tasks. Table 4.1 summarises the various performance metrics collected per task during a simulation. Once the simulation is complete, the simulator can visualize and process the performance metrics generating a comma-separated (CSV) file with an

adequate layout suitable to be imported into MS Excel or SPSS for further, more detailed analysis.

Metrics:
Metrics for all subtasks:
• Task completion time • Path length of the tip traversed during the task • Maximum and average force received by the endoscope's tip during the task • Maximum and average force received by the endoscope's tip section (last 3cm) • Maximum and average velocities and accelerations of the physical host • Maximum and average velocities and accelerations of the virtual endoscope • Maximum and average force-feedback generated by the haptic device • Number of movements performed • Movement properties (path length, duration, economy)
Metrics for the clipping and cutting subtasks:
• Clipping / cutting distance in centimetres from the indicated point • Clipping / cutting angle between the clipping / cutting tool and the surface • Number of clippings /cuttings • Degree of instrument protrusion during the operation
Metrics for the gallbladder dissection subtasks:
• Number of instances diathermy is activated • Total time diathermy is activated • Percentage of time burning non-target tissue

Table 4.1: Metrics collected per task during a simulation

For each completed task, an associated binary file containing the motions of the haptic device and of the virtual endoscope is also generated. From these files, a number of additional metrics can be extracted. A special tool was developed (Figure 4.16) that allows for visual comparison of these metrics between different attempts of the same tasks.

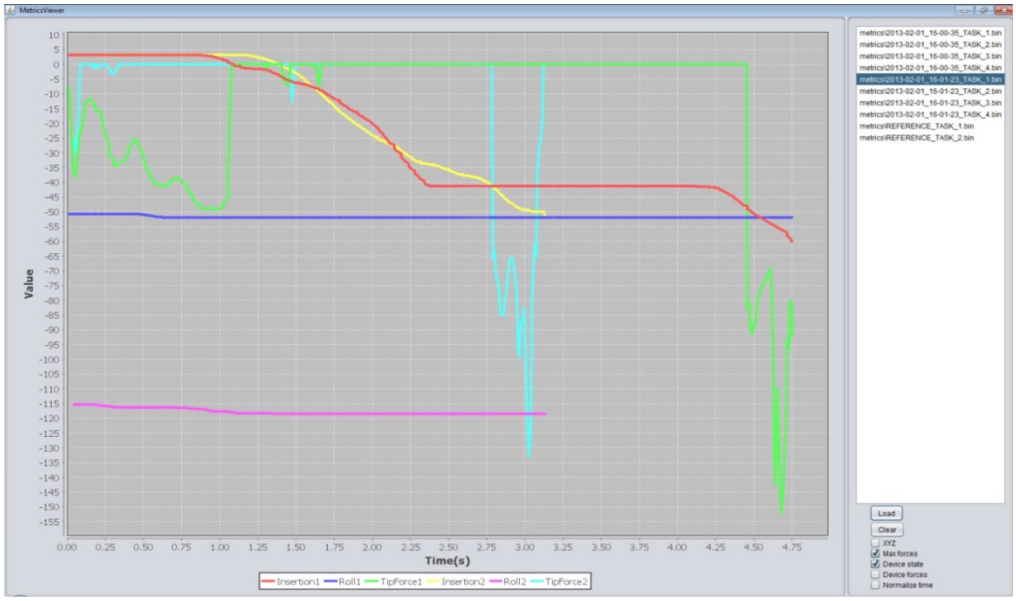


Figure 4.16: The metrics analysing tool. Visually compares two completions of the same task in terms of device insertion in cm (red and yellow), roll in angles (blue and magenta) and the force exerted on the tip (no units, green and cyan).

4.4.3 Validation

A validation study was designed and conducted with the assistance of Mr Dan Brown, an Academic Foundation Year doctor, in order to establish face validity (the degree to which the simulation mimics reality), content validity (the usefulness of the simulation as a training tool) and construct validity (the ability of the simulator to differentiate expert and novice users) for the simulator. The study received research and development approval by Imperial College London and was given a favourable ethical opinion by the Imperial College Research Ethics Committee (Ref: ICREC_13_4_4). Study documents are included in Appendix C.

Three groups of participants were recruited into the study. The first group comprised experts in NOTES surgery, defined by having performed ≥ 10 animal-model or human NOTES procedures. The second group comprised surgical trainees with experience of both endoscopy (≥ 10 of any combination of oesophagogastricdueodenoscopies / small bowel enteroscopies / colonoscopies / flexible sigmoidoscopies performed independently) and laparoscopic operations (≥ 10 procedures performed independently), but with no or little experience of

NOTES. The third group comprised gastroenterologists with experience of endoscopy (as outlined above), but with no or minimal experience of NOTES or laparoscopic operations. The reason for separating Gastroenterologists from Novices was to compare which prior experience, endoscopic or laparoscopic, has a bigger impact on acquisition of skills in NOTES.

The participants were required to complete three trans-gastric hybrid (i.e. with gallbladder retraction done by a laparoscopic grasper) NOTES cholecystectomies. Prior to performing their first procedure, all participants were given a technical instruction sheet (see Appendix C) outlining the nature of the simulation and were informed of what help they may receive from the researcher (who was acting as assisting surgeon) during the procedure (holding the endoscope in a particular position, activating the instruments and retracting the gallbladder). These actions are performed by an assisting surgeon in real life. In order to prevent bias, the researcher was only acting following a direct instruction for the operator. After reading the instruction sheet, participants were given a maximum of 3 minutes to familiarize themselves with basic navigation of the endoscope and how to operate the instruments in a non-anatomical “sandbox” environment (Figure 4.17).

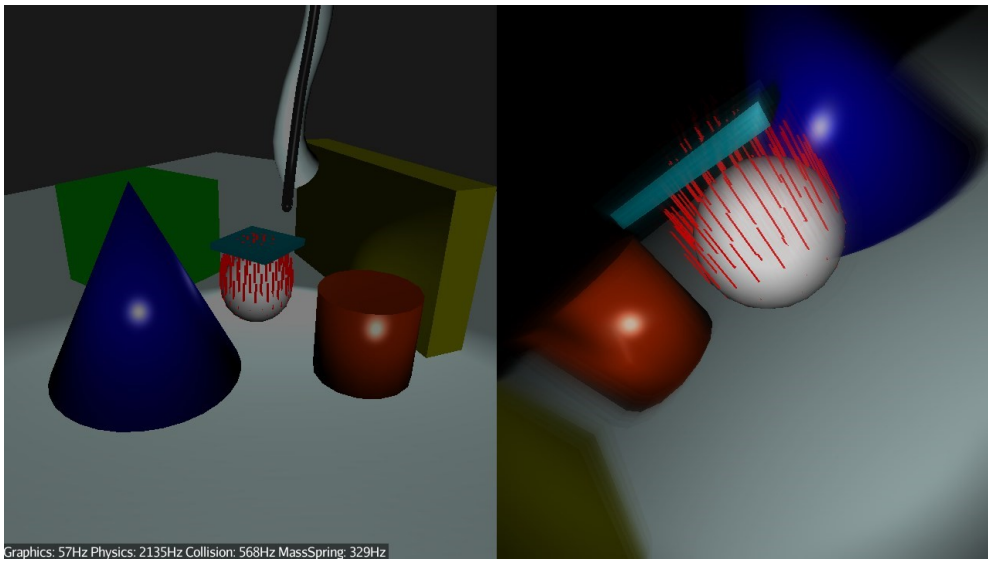


Figure 4.17: A sandbox environment used to familiarize the participants with basic endoscope navigation

Prior to commencing their first recorded procedure, participants were given the opportunity to ask questions relating to the practicalities of the simulation, but were not allowed to request any technical advice as to how best to perform the procedure. After completing all 3 procedures, the participants were given an online anonymous questionnaire (see Appendix C) to fill regarding their experience (year of training, number of procedures done in human/on simulators), face validity (graphical appearance, behaviour of tools and tissue, difficulty of the procedure, overall realism) and content validity (adequacy of the simulated tasks and perceived utility of the simulator as a training tool for NOTES). Construct validity was evaluated by comparing operative performance metrics such as operative speed, endoscope path length and instrumentation accuracy of subjects. Given the non-normality of the data, a two-tailed Mann–Whitney U test was used on each metric using statistical analysis software (SPSS by IBM, New York, USA).

4.4.3.1 Demographics

A total of 14 surgeons in different specialities (4 upper Gastro-intestinal, 4 lower Gastro-intestinal, 4 gastroenterology, 1 breast surgery and 1 unspecialized) participated in the study. Four of them were qualified as NOTES experts who have performed 10 or more human or animal-model NOTES procedures independently. This is analogous to previous validation studies of a similar design (Bittner et al., 2010, Jensen et al., 2013). Table 4.2 summarizes the demographics and specialization of the cohort.

	Group A (Experts)	**Group B (Novices)**	**Group C (Gastroenterologists)**
n	4	7	3
Age	35.5 (33-52)	34 (31-36)	33 (33-46)
Postgraduate year of training (PGY)*	9 (6-30)	7 (4-12)	9 (2-20)
Male	100%	100%	66.6%
Right handed	75%	100%	100%
Upper gastrointestinal surgeons	1	3	0
Lower gastrointestinal surgeons	1	3	0
Breast surgeons	0	1	0
Unspecialised	1	0	0
Gastroenterologists	1	0	3

*Table 4.2: Demographics. *Only years with >50% clinical practice included. Continuous values quoted as median with range in brackets.*

Table 4.3 presents the average number of procedures performed independently by each group.

	Group A (Experts)		**Group B (Novices)**		**Group C (Gastroenterologists)**	
	Animals or Simulators	Humans	Animals or Simulators	Humans	Animals or Simulators	Humans
Oesophagoduodeno gastroscopy	10 (0-25)	5 (0-80)	0 (0-55)	55 (5-150)	10 (5-55)	1000 (800-6000)
Small bowel enteroscopy	0 (0-0)	0 (0-0)	0 (0-0)	0 (0-0)	0 (0-5)	20 (0-200)
Colonoscopy	0.5 (0-20)	30 (0-20000)	0 (0-100)	10 (0-50)	0 (5-20)	800 (250-3000)
Flexible sigmoidoscopy	5 (0-20)	30 (0-3000)	0 (0-5)	20 (0-50)	0 (0-0)	1000 (200-1000)
Any endoscopic procedure	18 (0-60)	105 (0-23000)	10 (0-100)	75 (25-212)	30 (10-60)	2820 (1250-10200)
Any laparoscopic procedure	12.5 (0-200)	80 (0-350)	20 (2-120)	120 (0-250)	0 (0-0)	0 (0-0)
Any natural orifice transluminal endoscopic surgery (NOTES) procedure	12.5 (10-20)	1.5 (0-4)	0 (0-3)	0 (0-0)	0 (0-3)	0 (0-0)

Table 4.3: Number of procedures performed independently by participants

The majority of participants were interested in NOTES surgery and VR simulator for NOTES. The specific questions were:

Q1: I am interested in NOTES surgery

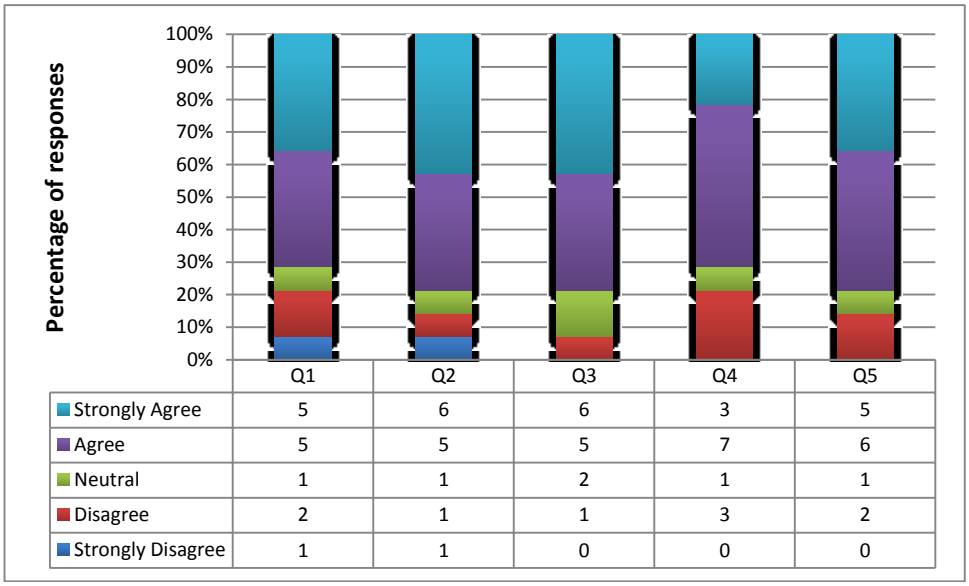
Q2: I am interested in the concept of NOTES virtual reality simulation

Q3: I am interested in the concept of virtual reality simulation in surgery in general

Q4: I believe NOTES virtual reality simulation could prove useful in surgical training curricula

Q5: I believe virtual reality simulation in general will prove useful in surgical training curricula

Table 4.4 presents the Likert scale responses to the respective questions. On average 76% out of all responses were “Agree” or “Strongly Agree” and 16% were “Disagree” or “Strongly Disagree”. The remaining 8% of responses were “Neutral”.



	Q1	Q2	Q3	Q4	Q5
Strongly Agree	5	6	6	3	5
Agree	5	5	5	7	6
Neutral	1	1	2	1	1
Disagree	2	1	1	3	2
Strongly Disagree	1	1	0	0	0

Table 4.4: Interest in VR simulator for NOTES - Liker scale responses

Table 4.5 summarizes the participants' previous experience with simulators. Only one person did not use any simulator before and nearly half (43%) used both laparoscopic and endoscopic simulators before. From the other half of participants who used just one type of simulator previously, the laparoscopic simulators were more popular (36%) than the endoscopic ones (14%).

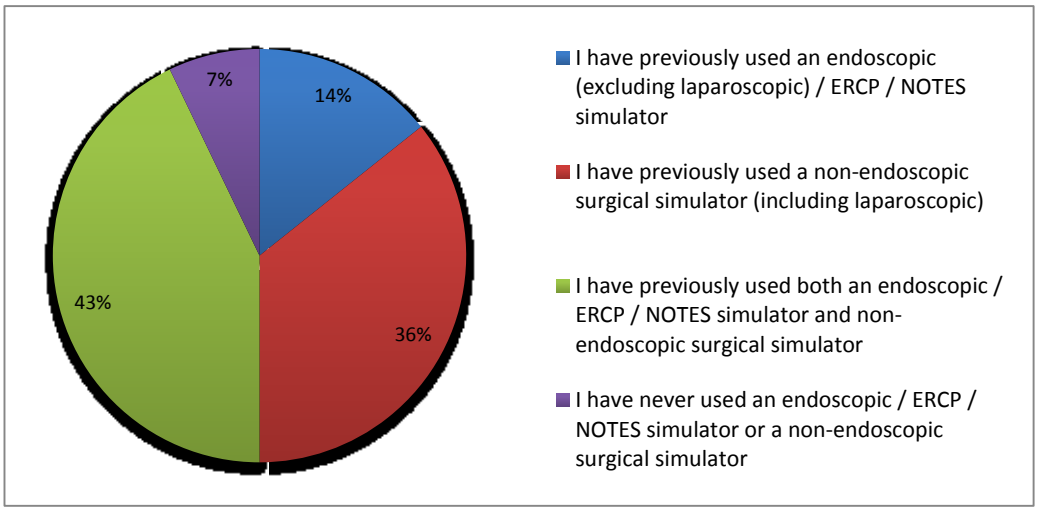


Table 4.5: Previous experience with simulators

4.4.3.2 Face validity

Face validity was evaluated by asking participants to complete the aforementioned questionnaire after completing their 3 procedures. The questionnaire assessed graphical appearance, behaviour of tools and tissue, difficulty of the procedure and overall realism. Specifically it contained 15 questions:

Q1: The endoscope clipper and scissors were visually realistic

Q2: The endoscope diathermy was visually realistic

Q3: The endoscope grasper was visually realistic

Q4: The tissues and organs were visually realistic

Q5: The endoscope hardware was visually realistic

Q6: The endoscope hardware felt realistic

Q7: The movement of the tip of the endoscope was realistic

Q8: The movement of the shaft of the endoscope was realistic

Q9: The amount and nature of 'looping' of the endoscope was realistic

Q10: The freedom of movement of the endoscope was realistic

Q11: The length of the endoscope was realistic

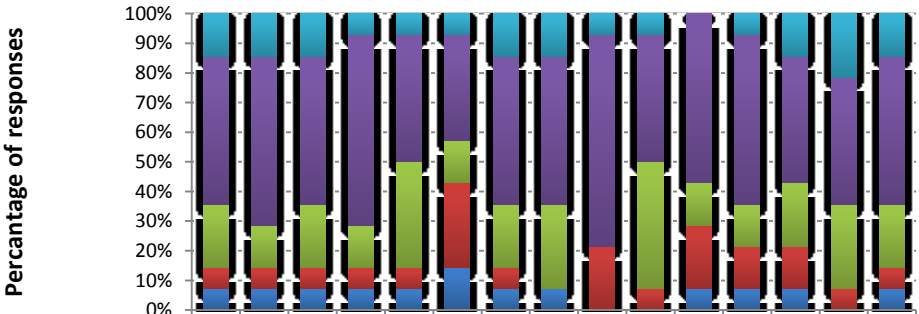
Q12: The interaction of the endoscope and instruments with the tissues was visually realistic

Q13: The interaction of the endoscope and instruments with the tissues felt realistic

Q14: The difficulty of the simulated procedure was realistic

Q15: Overall the simulator was realistic

Table 4.6 presents the participants responses.



	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Q11	Q12	Q13	Q14	Q15
Strongly Agree	2	2	2	1	1	1	2	2	1	1	0	1	2	3	2
Agree	7	8	7	9	6	5	7	7	10	6	8	8	6	6	7
Neutral	3	2	3	2	5	2	3	4	0	6	2	2	3	4	3
Disagree	1	1	1	1	1	4	1	0	3	1	3	2	2	1	1
Strongly Disagree	1	1	1	1	1	2	1	1	0	0	1	1	1	0	1

Table 4.6: Face validity - Likert scale responses

62% of responses to statements regarding the realism of the virtual endoscope behaviour were “agree” or “strongly agree”. The statements were: “The movement of the tip of the endoscope was realistic”, “The movement of the shaft of the endoscope was realistic”, “The amount and nature of 'looping' of the endoscope was realistic”, “The freedom of movement of the endoscope was realistic” and “The interactions of the endoscope and instruments with the tissues were realistic”.

67% of responses to statements regarding the visual realism of the virtual endoscopic camera view were “agree” or “strongly agree”. The statements were: “The endoscope clipper and scissors were visually realistic”, “The endoscope

diathermy was visually realistic", "The endoscope grasper was visually realistic", "The tissues and organs were visually realistic" and "The interactions of the endoscope and instruments with the tissues were visually realistic".

The participants were most critical on the statements pertaining to the haptic device ("The endoscope hardware felt realistic", "The endoscope hardware was visually realistic") with 29% of responses "disagreeing" or "strongly disagreeing", 25% being "neutral" and 46% "agreeing" or "strongly agreeing". Upon further exploration, this was found to be primarily caused by the design of the handpiece, which included force-feedback enabled thumb-wheels. This added to the complexity of the shape and weight of the handpiece, resulting in, as described by some participants, a "chunky" or "clumsy" feel.

The participants had an opportunity to give free text comments:

- *"All or any comments I could make would be petty as it is a very nice simulator. Well done! Perhaps the weight of control handle and passive torque on shaft could be lightened." (**Expert**).*
- *"The instrument is much too stiff and heavy." (**Expert**).*
- *"Scope very clumsy to handle. Far too long and stiff; however, movement of tip was realistic. Felt like positioning on the gallbladder for dissection was more luck then judgement but I am not used to the anatomy. (**Gastroenterologist**)."*
- *"Very unrealistic when compared to performing a laparoscopic cholecystectomy in a person. (**Novice**)."*
- *"Insertion length was short. Sometimes you can push through a loop a bit. (**Novice**)."*

Summarizing the face validity of NOViSE, in 14 out of 15 statements, ≥50% of responses were "agree" or "strongly agree". 9 out of 14 (64%) participants agreed or strongly agreed that the "The difficulty of the simulated procedure was realistic"

and “Overall the simulator was realistic”. It is also worth noting that a total of 11 out of 14 “strongly disagree” responses came from the same novice participant.

4.4.3.3 Content validity

Content validity was evaluated by asking participants to complete the aforementioned questionnaire after completing their 3 procedures. The questionnaire assesses the adequacy of the simulated tasks and perceived utility of the simulator as a training tool for NOTES (Table 4.7). Specifically, it contained 6 questions:

Q1: Tasks 1-3 (Navigation of the endoscope into the peritoneal cavity) are useful training tool for NOTES

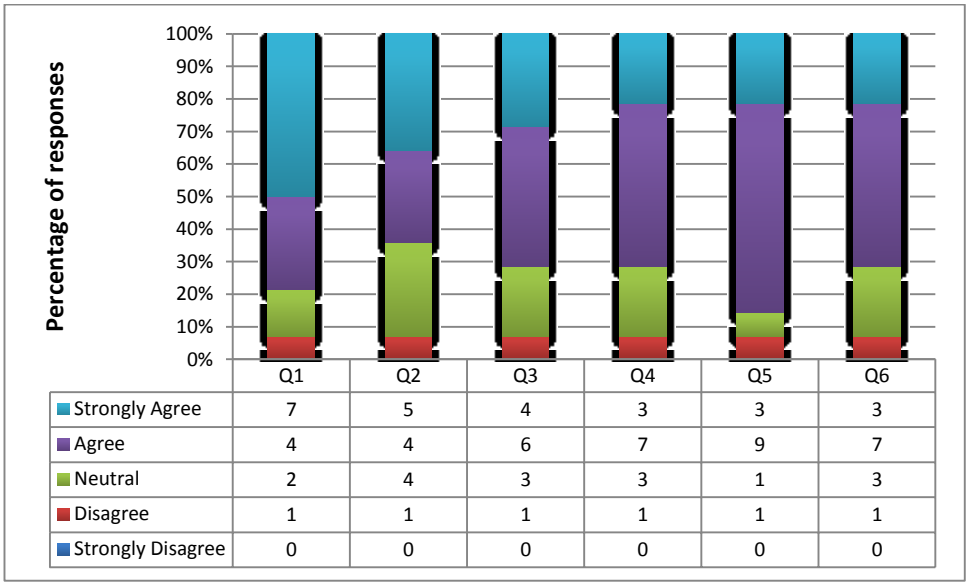
Q2: Tasks 4-9 (Clipping and cutting the artery and the duct) are useful training tool for NOTES surgery

Q3: Task 10 (Dissecting gallbladder from the liver bed) is a useful training tool for NOTES surgery

Q4: The range of exercises provided by the simulator is sufficient to make it a useful training tool for NOTES

Q5: Overall the simulator is a useful training tool for NOTES surgery

Q6: I would recommend the simulator to others



	Q1	Q2	Q3	Q4	Q5	Q6
Strongly Agree	7	5	4	3	3	3
Agree	4	4	6	7	9	7
Neutral	2	4	3	3	1	3
Disagree	1	1	1	1	1	1
Strongly Disagree	0	0	0	0	0	0

Table 4.7: Content validity - Likert scale responses

In terms of content validity, 71.4% responses to the statements assessing the usefulness of the individual tasks (navigation into the peritoneal cavity, clipping and cutting the artery and the duct, and dissecting the gallbladder from the liver bed) as a training tool were “agree” or “strongly agree”.

10 out of 14 (71.4%) participants “agreed” or “strongly agreed” that “The range of exercises provided by the simulator is sufficient to make it a useful training tool for NOTES surgery” and would recommend the simulator to others. 12 out of 14 (85.7%) stated that overall the simulator is a useful training tool for NOTES surgery.

The free text responses were as follows:

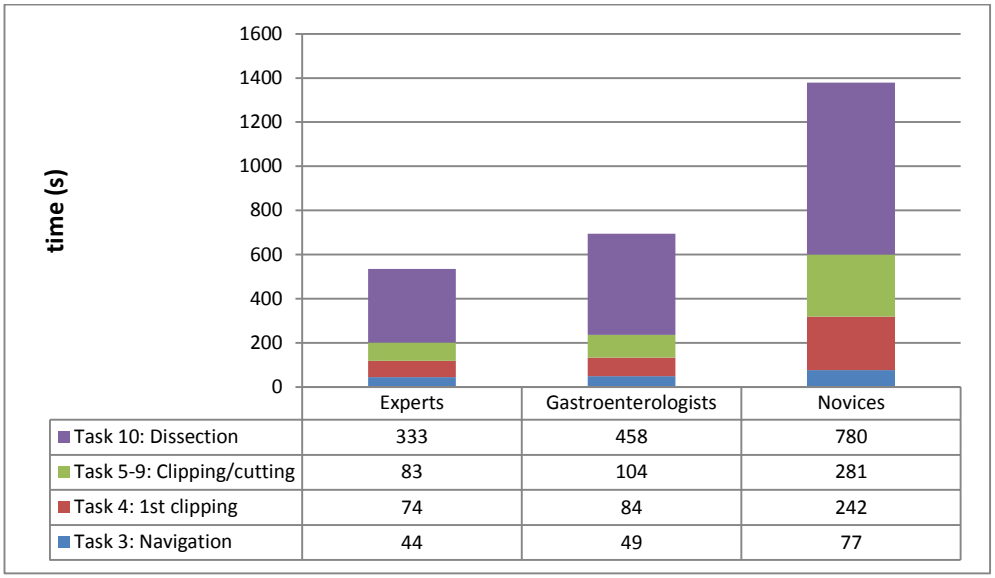
- *“No current secure clip available for endoscope. Current transgastric cholecystectomy performed using fundal first technique then endoloops to CD.” (**Expert**).*
- *“This simulator is still a work in progress so therefore is difficult to comment on whether or not this is a good training tool. As I have not performed NOTES surgery I cannot comment on its effectiveness as a training tool. Intuitively any simulator should help with real world surgery - but this is dependent on the fidelity and responsiveness of the simulator.” (**Novice**).*
- *“Refinement required with clipping and cutting.” (**Novice**).*
- *“Not sure if realistic or not as have never done a NOTES cholecystectomy, but feel scope needs to be easier to handle so very fine movements can be practiced.” (**Gastroenterologist**).*

Summarizing the content validity of NOViSE, 30% of responses concerning the content validity were “strongly agree”, 44% “agree”, 19% were “neutral” and 7% “disagree”. None of the participants “strongly disagreed” with any content validity statement.

Overall, the results of the face and content validity study range mainly from positive to very positive, indicating that the NOViSE simulator authentically recreates a trans-gastric hybrid cholecystectomy operation and that it has potential as a useful training tool for NOTES surgery.

4.4.3.4 Construct validity

Operative metrics are shown in Figure 4.18 – 4.20 and Table 4.8 - 4.12. P-values smaller than 0.05 ($p < 0.05$) was observed for two performance metrics. However, due to small sample size ($n = 14$), a statistically significant difference cannot be stated. Instead, a trend indicating that this is likely can be concluded. Experts were faster and navigated using shorter paths than novices in all but the first task which involved navigation from the oesophagus via the stomach into the peritoneal cavity. This task required only basic navigation skills, which might explain why no trend was demonstrated. All three following tasks indicated such a difference. This is because the navigation in an open space of the abdomen, without the lumen supporting the endoscope, is considerably more challenging. The unexperienced operators were often not aware of the looping and tension of the not visible endoscope shaft. Some novices were complaining that the tip was not moving or, the opposite, moving by itself, without their action. This was caused by looping and tension of the shaft accumulated at one point and released at another. Additionally, access to the gallbladder, especially to the artery and duct, required bending the tip almost 180 degrees resulting in an “inverted” behaviour. Pushing the endoscope in was resulting in moving away from the point of interest and vice versa. This apparently simple dependence turned out to be confusing for some participants.



	Experts	Gastroenterologists	Novices
Task 10: Dissection	333	458	780
Task 5-9: Clipping/cutting	83	104	281
Task 4: 1st clipping	74	84	242
Task 3: Navigation	44	49	77

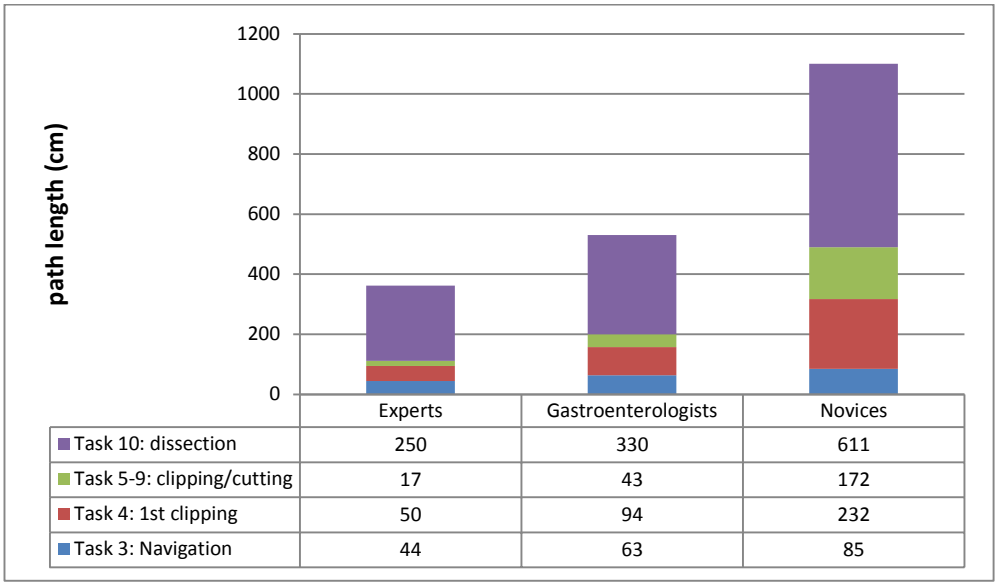
Figure 4.18: Task completion times

	Experts v Novices P-Value	Experts v Gastroent. P-Value	Novices v Gastrornt. P-Value
Task 3: Time taken to exit stomach (s)	**0.32**	**1.00**	**0.67**
Task 4: Time from exiting stomach to application of first staple (s)	**0.01**	**0.63**	**0.07**
Tasks 5-9: Time from application of first staple to application of last staple (s)	**0.02**	**0.40**	**0.19**
Task 10: Time taken to dissect gallbladder from liverbed (s)	**0.02**	**0.23**	**0.27**

Table 4.8: Task completion times

During cystic artery and duct clipping and cutting, experts were clearly more aware of how to position the endoscope tip to be able to efficiently clip and cut the designated points just by using the thumbwheels. Novices had to often reposition the whole endoscope, which took considerably more time and lengthened the paths. The assisting investigators noted that, on average, experts were more likely to ask for their help. Mainly, to hold the physical shaft while there was high torque so they could precisely steer the tip and protrude/intrude actuators using both hands. To finish the clipping and cutting the cystic artery and duct, experts on average needed just over one minute, gastroenterologists over two minutes and

novices nearly 7 minutes in total. Both time and endoscope path during deploying the first clip, as well as the subsequent clippings and cutting were significantly shorter among the experts than novices (p = 0.01). The deviation from the optimal 90 degrees clipping and cutting angle (Figure 4.20 and Table 4.10) was around 20 degrees for experts and novices and 16.4 degrees for gastroenterologists.



	Experts	Gastroenterologists	Novices
Task 10: dissection	250	330	611
Task 5-9: clipping/cutting	17	43	172
Task 4: 1st clipping	50	94	232
Task 3: Navigation	44	63	85

Figure 4.19: Path lengths

	Experts v Novices P-Value	Experts v Gastroent. P-Value	Novices v Gastrornt. P-Value
Task 3: Endoscope path length prior to exiting stomach (cm)	**0.16**	**0.63**	**0.84**
Task 4: Endoscope path length from exiting stomach to application of first staple (cm)	**0.01**	**0.40**	**0.27**
Tasks 5-9: Endoscope path length from application of first staple to dissection of cystic duct (cm)	**0.01**	**0.63**	**0.18**
Task 10: Endoscope path length during gallbladder dissection from liverbed (cm)	**0.02**	**0.40**	**0.12**

Table 4.9: Path lengths p-values

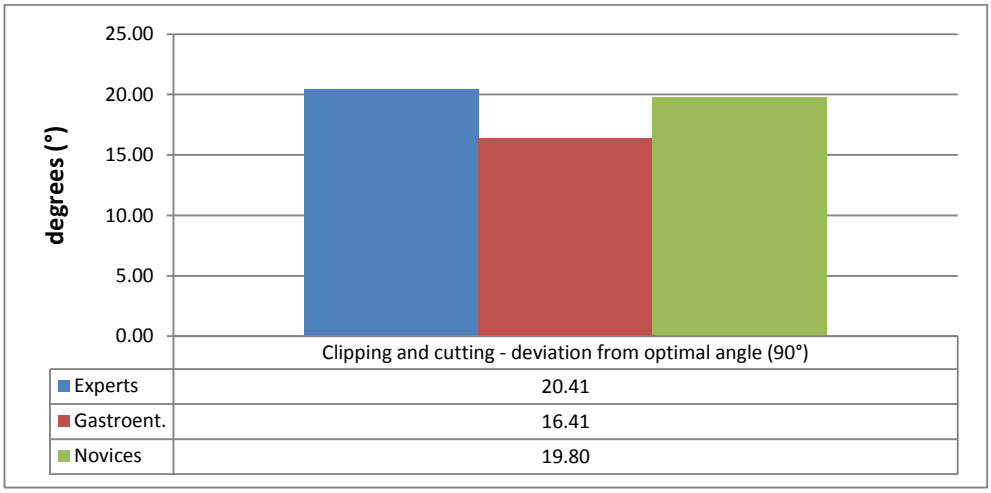
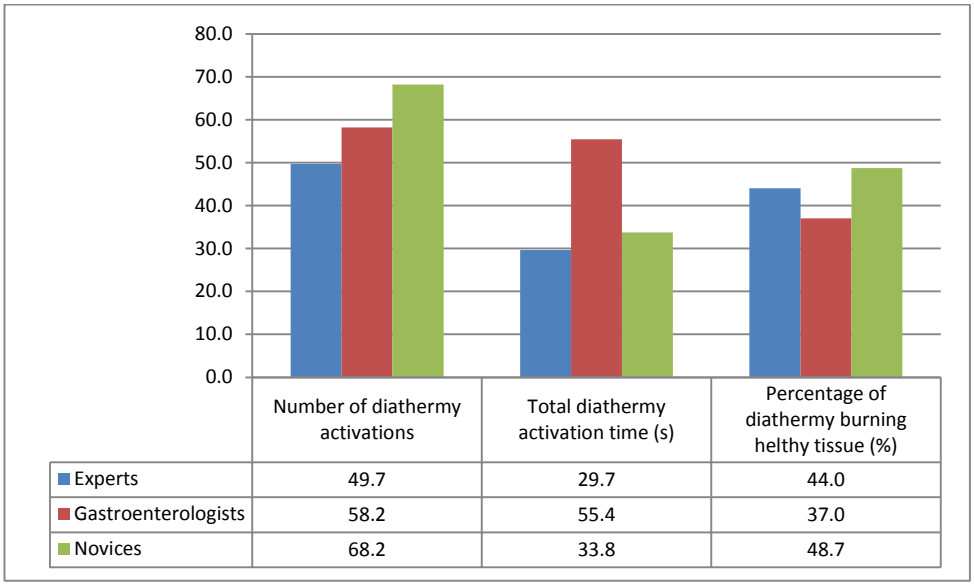


Figure 4.20: Clipping and cutting - deviation from optimal angle (90 deg.)

This was also the case during the gallbladder dissection using the diathermy tool. Experts were performing more work just by steering the tip using the thumbwheels. This, perhaps surprisingly, did not result in improved dissection accuracy ($p = 0.32$, Figure 4.21 and Table 4.10), but shortened the task completion time by more than half from 13 to 5.5 minutes ($p = 0.02$). These findings may be attributed to the fact that novices were more cautious to avoid damage to healthy tissue than more experienced colleagues who knew what extent of damage is acceptable in order to quickly perform the task.



	Number of diathermy activations	Total diathermy activation time (s)	Percentage of diathermy burning helthy tissue (%)
Experts	49.7	29.7	44.0
Gastroenterologists	58.2	55.4	37.0
Novices	68.2	33.8	48.7

Figure 4.21: Diathermy use

	Experts v Novices P-Value	Experts v Gastroent. P-Value	Novices v Gastrornt. P-Value
Number of diathermy activations	**0.11**	**0.23**	**0.27**
Total diathermy activation time	**0.79**	**0.06**	**0.18**
Percentage of diathermy activation on healthy tissue	**0.32**	**0.40**	**0.18**
Average deviation of clip from the ideal 90 degrees in relation to cystic artery / duct	**0.93**	**0.15**	**0.52**

Table 4.10: Diathermy use and clipping / cutting p-values

Novices found efficient navigation in the open cavities challenging. This was clearly reflected by their increased operative time and endoscope path length compared to the experts. However, there was no statistically significant difference between experts and novices in the remaining gathered metrics (Table 4.11). Particularly, in the velocities and accelerations that they used to manipulate the endoscope and in the resulting force feedback. This suggests that novices knew how to manipulate the endoscope safely, although not as efficiently as experts.

	Group A (Experts)	Group B (Novices)	Group C (Gastroenterologists)	Experts vs Novices p-values	Experts vs Gastroen. p-values	Novices vs Gastroen. p-values
Operative forces:						
Average force applied to tissue by tip of endoscope	29.7	33.9	32.5	0.23	0.40	1.00
Maximum force applied to tissue by tip of endoscope	4834	9728	6271	0.93	0.63	0.84
Average force feedback (N)	0.04	0.01	0.05	0.93	0.40	0.27
Maximum force feedback (N)	2.04	4.12	3.46	0.16	0.40	1.00
Clipping and cutting:						
Average deviation of clip from the ideal 90 degrees in relation to cystic artery / duct	20.4	19.8	16.4	0.93	0.15	0.52
Average number of clippings/cuttings per task	1.32	1.58	1.31	0.31	0.86	0.52
Distance from clipping/cutting point to indicated one (cm)	2.5	2.2	2.3	0.16	0.63	1.00
Clipper/cutter protrusion	8.2	9.1	8.8	0.23	0.63	1.00
Endoscope velocities and accelerations:						
Average endoscope shaft velocity (m/s)	0.003	0.003	0.005	0.65	0.23	0.07
Maximum endoscope shaft velocity (m/s)	0.144	0.187	0.161	0.79	0.86	0.27
Average endoscope shaft acceleration (m/s^2)	0.014	0.008	0.019	0.53	0.63	0.27
Maximum endoscope shaft acceleration (m/s^2)	0.917	1.275	1.051	0.16	0.63	0.27

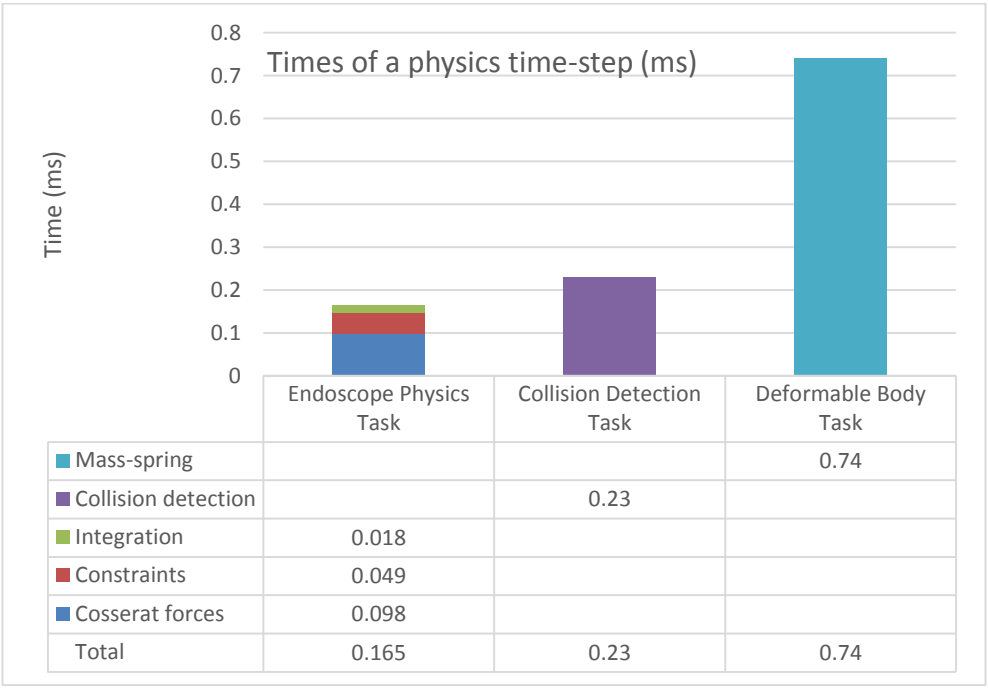
Table 4.11: The remaining metrics with the corresponding p-values.

Summarizing, construct validity indicates a trend for the following simulator metrics for Experts vs Novices test: time from exiting stomach to application of first

staple (74.0s vs 241.7s, p = 0.01), time from application of first staple to application of last staple (82.9s vs 281.16s, p = 0.02), time from application of last staple to completed dissection of gallbladder from liver bed (333.4s vs 779.8s, p = 0.02), endoscope path length from exiting stomach to application of first staple (50.4cm vs 232.4cm, p = 0.01), endoscope path length from application of first staple to dissection of cystic duct (16.8cm vs 172.4cm, p = 0.01), endoscope path length from dissection of cystic duct to complete dissection of gallbladder from liver bed (250.4cm vs 611.1cm, p = 0.02). Construct validity was not demonstrated for the remaining metrics and groups, i.e. experts vs gastroenterologists and novices vs gastroenterologists. However, a trend can be observed indicating that the gastroenterologists were faster and used shorter path lengths than novices in all tasks. This does correlate with the findings of (Nehme et al., 2013) who found prior endoscopic experience to be of greater benefit than prior laparoscopic experience in acquiring skills in NOTES.

4.4.4 Computational performance

The computational performance during the procedure was tested on an Asus N55s laptop (Win7 x64, Intel Core i7 2.2 GHz, 8GB RAM, NVidia GeForce GT 555M). Table 4.12 presents average computational times during the simulated NOTES procedure for the particular simulation (sub)-tasks: endoscope physics, collision detection and deformable body simulation.



	Endoscope Physics Task	Collision Detection Task	Deformable Body Task
Mass-spring			0.74
Collision detection		0.23	
Integration	0.018		
Constraints	0.049		
Cosserat forces	0.098		
Total	0.165	0.23	0.74

Table 4.12: The computational performance of NOViSE.

The total computational time of virtual endoscope (Cosserat forces, constraints, integration) consisting of 100 Cosserat elements was below 0.2ms. The collision detection runs in sync on a separate thread and slightly slows down the rod physics (0.23ms). Although it stays one step behind, no problems related to this fact were observed. The mass-spring model of the deformable gallbladder was the slowest part of the simulation requiring nearly 0.75ms per update. The display was updated at a standard 60 frames per second.

4.5 SUMMARY

This chapter has presented the first prototype virtual reality simulator for NOTES which supports a complete trans-gastric hybrid cholecystectomy procedure using a flexible endoscope. The behaviour of the virtual flexible endoscope is based on a modified implementation of the CoRdE model presented in Chapter 3. The model allows for the realistic recreation of all deformations of the endoscope, such as bending and twisting, as well as guaranteeing a fast, real-time response to user manipulations. The physical parameters of the virtual endoscope were manually adjusted to match the behaviour of the real endoscope. There are four types of actuators available to the operator: grasper, clipper, scissors and diathermy tool. The operator interacts with the virtual endoscope via a custom built haptic device. The highly-optimized implementation allows the endoscope simulation to run efficiently on an off-the-shelf PC or laptop significantly exceeding haptic interactive rates (4 kHz).

NOViSE showed good overall face and content validity in the study. Participants indicated that the looping phenomenon of the endoscope was the most realistic aspect of the simulation. They mainly complained about the feel of the endoscope hardware, stating that the physical shaft was too stiff and heavy, and a lack of ergonomic design of the handpiece. In terms of content validity, participants largely agreed that NOViSE is a useful training tool for NOTES and that they would recommend it to others. Regarding construct validity, a trend was observed that experts were faster and used a shorter endoscopic path length than novices in all but the first task. The remaining metrics did not demonstrate statistically significant difference between the experts and novices. There were also no significant differences between experts and gastroenterologists, or between novices and gastroenterologists.

The main limitations of NOViSE are that it currently only simulates transgastric hybrid cholecystectomy and owing to the fact that it is a prototype, some features of the hardware, such as the endoscope's length and visual realism, as well as the

use of foot pedals to activate the clipper and scissors, are known to be sub-optimal. The same is true of certain aspects of the computer simulation, such as the absence of fat in Calot's triangle.

The validation study suffered from small numbers in each of the participant groups, hence only trends could be concluded. There was also a large variation in procedural experience within each group. Participants completed their procedures at times convenient to them, which meant that some participants performed all three procedures in sequence, whilst others performed single procedures separated by several days or weeks. This may have influenced the results depending on whether repeated operating led to fatigue or greater familiarity with the procedure.

Chapter 5

VCSIM3 – A VR SIMULATOR FOR CARDIOVASCULAR INTERVENTIONS

This chapter presents VCSim3 - a Virtual Reality simulator for cardiovascular interventions. Effective and safe performance of cardiovascular interventions requires excellent catheter / guidewire manipulation skills. These skills are currently mainly gained through an apprenticeship on real patients, which may not be safe or cost-effective. Computer simulation offers an alternative for core skills training. However, replicating the physical behaviour of real instruments navigated through blood vessels is a challenging task.

The inextensible Cosserat rod introduced in Chapter 3 is used to model virtual catheters and guidewires. First, the simulator is overviewed in Section 5.3.1. Section 5.3.2 extends the constraints framework to support interactions between instruments. Next, in Section 5.3.3, the mechanical parameters of six guidewires and three catheters are optimized with respect to their real counterparts scanned in a silicone phantom using CT. Section 5.3.4 describes supporting solutions such as fluoroscopic visualization, contrast flow propagation, cardiac motion, balloon inflation and stent deployment, which enable performing a complete angioplasty procedure. The results of Section 5.4.1 assess the accuracy of the proposed virtual instruments in comparison to real ones during navigation in a silicone phantom. Section 5.4.2 analyses the compressibility of the model, whilst Section 5.4.3 presents the detailed computational performance of the instruments. Section 5.4.5 demonstrates the ability of the simulator to support a complete angioplasty procedure. Lastly, Section 5.4.6 gives the results of a preliminary face and content validation study.

5.1 CARDIOVASCULAR INTERVENTIONS

Cardiovascular diseases (CVD) are the main cause of death in the developed world (WHO, 2011). Minimally invasive endovascular procedures, widely adopted in diagnosis and treatment of CVDs, improve recovery time, reduce patient trauma and health-care costs. During such procedures, endovascular clinicians insert long, thin, flexible surgical instruments – catheters and guidewires, into the patient's vascular system. Guided by medical imaging, they then navigate the catheter / guidewire pair into the coronary arteries to treat the pathology (Figure 5.1). An effective and safe performance of these procedures requires excellent instrument manipulation skills, which are still mainly gained through an apprenticeship on real patients. Drawbacks of the apprenticeship model include costs, reduced training opportunities and patient-safety (Bridges and Diamond, 1999). One possible alternative is training on computer-based, virtual reality (VR) simulators (de Montbrun and Macrae, 2012). The last decade has seen growing interest in the benefits of using VR medical simulators in a range of specialties, including endovascular interventions.

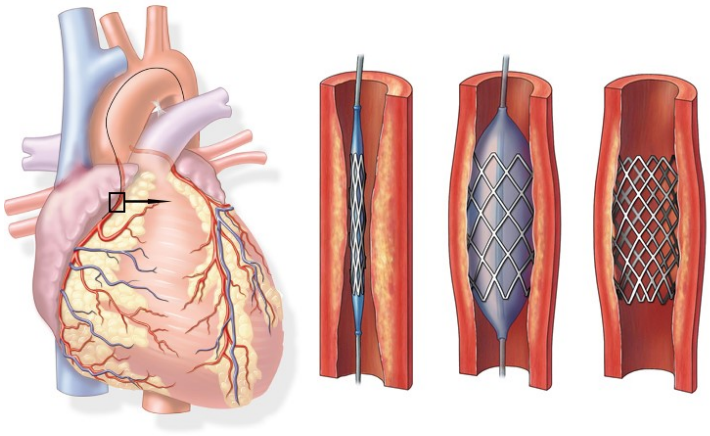


Figure 5.1: Coronary angioplasty procedure with stenting. (http://drbcshah.com)

5.2 RELATED WORK

Commercial VR vascular simulators such as VIST (*www.mentice.com*) or Angio Mentor (*www.simbionix.com*) have demonstrated a degree of face and content validity, but the ultimate realism is yet to be achieved (Gould et al., 2009). The fundamental part of such simulators is an underlying mathematical model of the virtual catheter and guidewire.

(Dawson et al., 2000) proposed a catheter model based on a multi-body system composed of rigid bodies and joints, which requires many links in order recreate a high degree of flexibility. (Wang et al., 2007) and (Luboz et al., 2009a) showed the possibility of simulating an elastic rod in real-time and with visually correct accuracy using a MSS. (Wang et al., 2007) recreated the material twist in MSS using a scalar torsion angle parameter.

Thanks to increasing computational power, solutions based on continuum mechanics approaches such as the finite element method (FEM) have become feasible. (Nowinski and Chui, 2001) applied a linear elasticity finite element representation, which assumes that the instruments move only with small displacements. (Duriez et al., 2006) introduced a static non-linear deformable beam model resulting in an accurate simulation of bending and twist deformations, whilst (Lenoir et al., 2006) applied this approach to simulate interactions between catheter and guidewire by modulating material property of the FEM model. (Alderliesten et al., 2004, Alderliesten et al., 2007) simulated rods, including friction, as a set of straight, non-bendable, incompressible beams with perfect torque control using a quasi-static approach. Later, (Li et al., 2011) improved this approach by using a FEM-based numerical solver.

More recently, (Duratti et al., 2008) applied a solution resembling the CoRde model (Spillmann and Teschner, 2007) to real-time interventional radiology simulation and (Tang et al., 2010, Tang et al., 2012) adapted the approach in (Bergou et al., 2008) to simulate catheters and guidewires.

5.3 METHODS

5.3.1 Simulator overview

The VCSim3 set-up (Figure 5.2) consists of a real-time simulation software, fluoroscopic view control and a VSP haptic-device (*Vascular Simulation Platform*, www.mentice.com). The software is written in Java and can efficiently run, i.e. exceeding haptic interactive rates, on a modern mid-range PC or laptop with Windows, MacOS or Linux operating systems. It is responsible for x-ray visualization and simulation of the virtual catheter/guidewire pair, interactions between them and vessels walls, contrast medium propagation, balloon inflation and stent deployment.

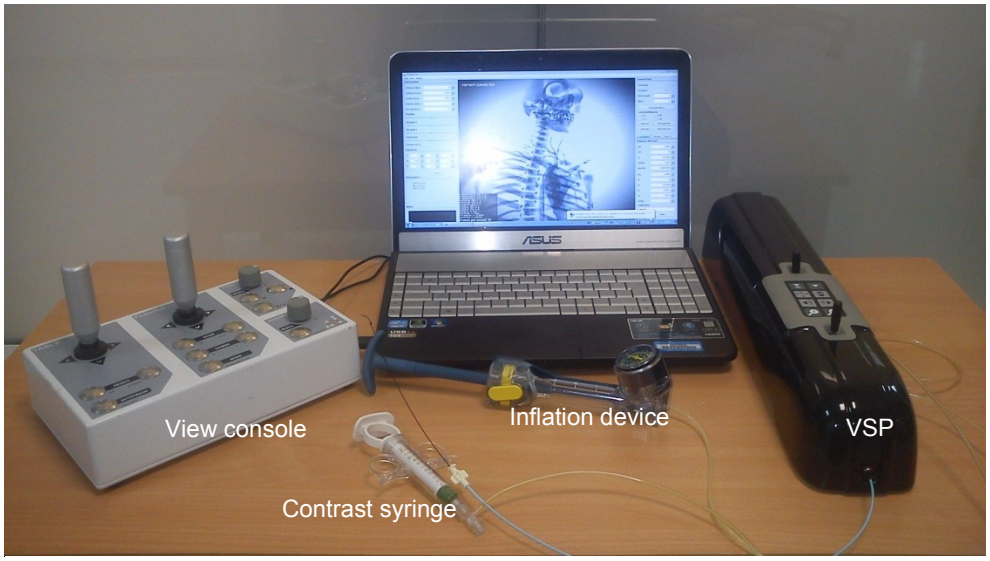


Figure 5.2: VCSim3 complete set-up including the simulator software running on the laptop, VSP haptic device, fluoroscopic view console, balloon inflation device, contrast injection syringe and fluoroscopy controls.

The operator controls the virtual catheters and guidewires using the VSP haptic device connected through a USB port. The VSP can track real endovascular instruments, however, it needs to be calibrated for particular diameters prior to use. The instruments can be pushed, pulled and rotated. Inside (Figure 5.3), the VSP is fitted with two optical sensors for instrument tracking and two force feedback mechanisms. The first sensor (b), placed near to the insertion slot (a), tracks the

catheter. The second one (d), located approximately 30cm further, tracks the guidewire. This results in a non-constrained guidewire tracking length, but limits the effective catheter tracking to 30cm. The force-feedback is generated by the motors (c, e) simply clamping the instruments. The VSP is additionally equipped with pressure and flow sensors. The former is used to connect a balloon inflation device and the latter to connect a syringe for contrast injection.

Figure 5.3: The VSP with removed chassis. Photography by Mr. Hafiz Harun.

An optional fluoroscopic view console (Figure 5.2) adapted from the VIST simulator (www.mentice.com) is equipped with two joysticks allowing for panning and rotating the x-ray view, as well as a set of buttons for controlling zooming and gamma. The view manipulations may also be done using the mouse and keyboard.

5.3.2 Virtual catheter and guidewire tool models

The catheter and guidewire models are based on the inextensible Cosserat Rod model introduced in Chapter 3. The focus was on real-time performance and realistic behaviour of tools and their interaction. In order to recreate the interactions between the instruments, a new constraint was added to the block iterative solver – the binding constraint. It is effectively a distance constraint with rest length set to zero aiming at keeping the guidewire inside the catheter. It

generates impulses parallel to the instruments centrelines between all the mass-points of the guidewire inside the catheter and the nearest two corresponding points on the catheter. These two points receive a factor of the impulse proportional to the distance from the guidewire point. The binding constraints are applied last after the distance and collision constraints.

The tips of instruments can have different material parameters. This allows for modelling of softer tips, which are common for many real instruments and prevent them from perforating or piercing the vessel walls. In order to keep the number of parameters manageable for the tip, the same parameters as for the shaft were used, but E_b is divided by a constant α. The intrinsic bend parameter $\hat{u}_1$ was used to model the curved tips of the instruments. Figure 5.4 depicts the above solutions in the case where a stiff guidewire straightens a soft curved tip of a catheter, and the opposite case where a soft guidewire follows a stiffer catheter.

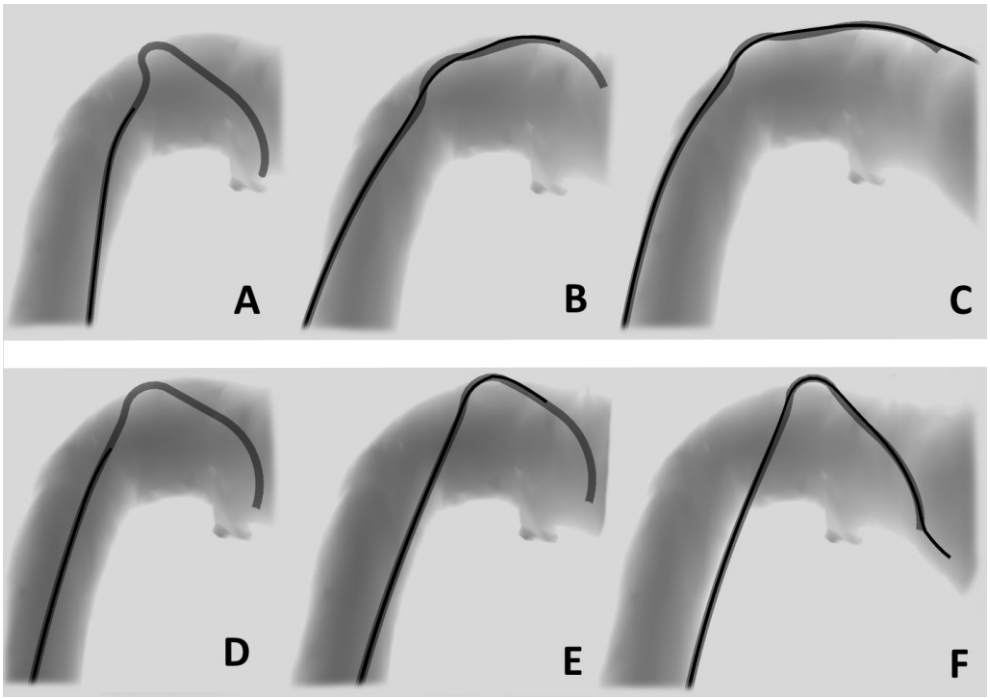


Figure 5.4: The interactions between a catheter and a guidewire. A, B, C (top row) show straightening of a soft catheter tip by a stiffer guidewire. D, E, F (bottom row) show a soft guidewire following a stiffer catheter.

5.3.3 Parameter optimization

Six real guidewires and three catheters were recreated by optimizing their parameters with respect to their real counterparts scanned in a silicone phantom model using CT. This data was acquired for the purposes of research and was first presented in (Luboz et al., 2011). The optimization routine in MatLab was implemented in collaboration with a visiting PhD student, Mr. Francisco Martínez-Martínez under joint supervision of Dr Bello and myself. The mechanical parameter optimization took into account the requirements for real-time usability and stability

In total, nine commonly used instruments made by Cook Medical Inc., Boston Scientific and Terumo Corp. were chosen for the simulation: three access guidewires (Cook Fixed Core Straight, Cook T-J-curved and Boston Bentson); a selection guidewire (Terumo Stiff Angled); two exchange guidewires (Boston Amplatz Super Stiff and Cook Rosen-Curved); and three diagnostic catheters (5F Beacon, Terumo 4Fr and Terumo 5Fr). Each instrument was inserted by the same operator at room temperature into a silicone vascular phantom (*Elastrat*). The set-up was scanned with the instruments reaching three different anatomical locations (Figure 5.5): common iliac artery bifurcation, aortic bifurcation and left renal artery origin. The scanner was a multi-detector CT with a resolution of 0.53x0.53x1 mm3. The 3D geometry of the phantom and inserted instruments was reconstructed using the snake segmentation algorithm in ITK-Snap. The instrument geometries were further processed to obtain their reference centrelines as showed in Figure 5.5 .

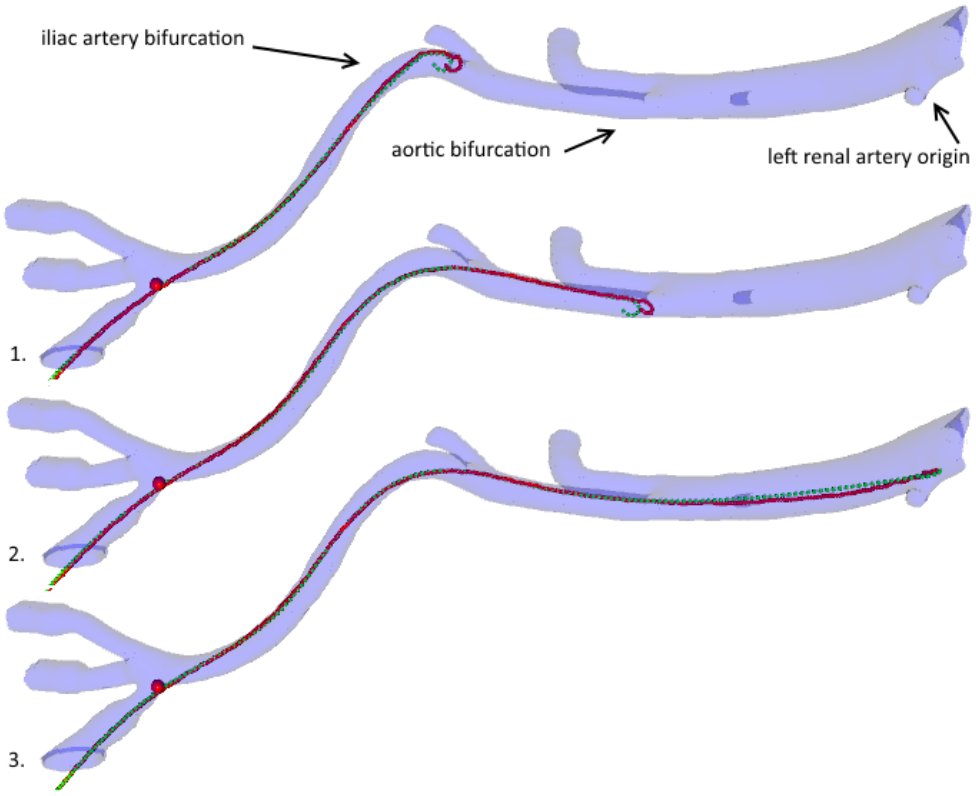


Figure 5.5: Reconstructed 3D geometry of the phantom showing the centreline of the Cook J guidewire in red and the simulated instrument centreline in green.

The virtual instruments were then automatically inserted into the virtual phantom. To ensure that the obtained parameters are valid for the required real-time performance, the physics simulation rate was capped at a steady rate significantly exceeding requirements for haptic interactivity (>1 kHz).

Matlab's Genetic Algorithm Toolbox (Chipperfield et al., 1994) was used to find optimal parameters of the Cosserat rod model. A similar method was previously applied in (Martinez-Martinez et al., 2013) to find optimal biomechanical parameters of human liver. The population size was set to 50 individuals and the algorithm stopped after 50 generations. The optimization took close to 2500 simulations per rod to converge (approx. 2 days). Each simulation involved automated insertion of the rod to the two first error measurement locations: iliac artery bifurcation and aortic bifurcation. The optimization consisted of the minimization of an error function calculated as the root mean square (RMS)

distance between the simulated mass-points and nearest points on the reference centreline, dependent on the following model parameters*:* Young modulus (E_b), radius (*r*), density *(d)*, spring constant of the parallel constraint (K_p) and the ratio between the Young modulus of the tip and the shaft (*α)*, which modelled the floppy nature of the tips of the instruments. The error function was an average RMS distance between simulated and reference centrelines:

$$\mathrm{f}\left(E_b, r, d, K_p, \alpha\right) = \sqrt{\frac{1}{N}\sum_{\mathrm{i}=1}^{N}(\mathrm{minDist}(\mathrm{p}_\mathrm{i}^\mathrm{s}))^2} \tag{19}$$

where *N* is the number of simulated mass-points inserted into the phantom and the function $\mathrm{minDist}(\mathrm{p}_\mathrm{i}^\mathrm{s})$ returns the nearest Euclidean distance to the reference centreline from the position of the *i*-th simulated mass-point – $\mathrm{p}_\mathrm{i}^\mathrm{s}$.

Due to availability of just one scanned dataset, the optimal parameters were obtained using the average RMS error value at the iliac artery bifurcation and aortic bifurcation. Next, they were validated at the deepest insertion point reaching the left renal artery origin. In the case of instruments with curved tips, the orientation aligning the virtual and reference tips in the same plane was manually obtained before parameter optimization, and used during automated insertion to rotate the virtual instrument tip between the three anatomical locations to the correct orientation.

5.3.4 Supporting solutions

In order for any cardiovascular simulator to be useful as a training tool, in addition to a realistic instrument model and haptic interface, it also needs to provide features such as x-ray visualization, cardiac motion, contrast medium propagation, balloon inflation and stent deployment.

5.3.4.1 Fluoroscopic visualization

The simulated fluoroscopy image (Figure 5.6) was produced using the open-source JME3 graphics engine. A custom X-Ray shader in GLSL was implemented which scales the surface opacity based on the surface normal vector and the view

direction, with noise effect at a post-processing stage. Off-the-shelf 3D models of the skeleton and an animated heart (*www.3dscience.com*) were used. The vessels and aorta were manually segmented from real patient CT scans.

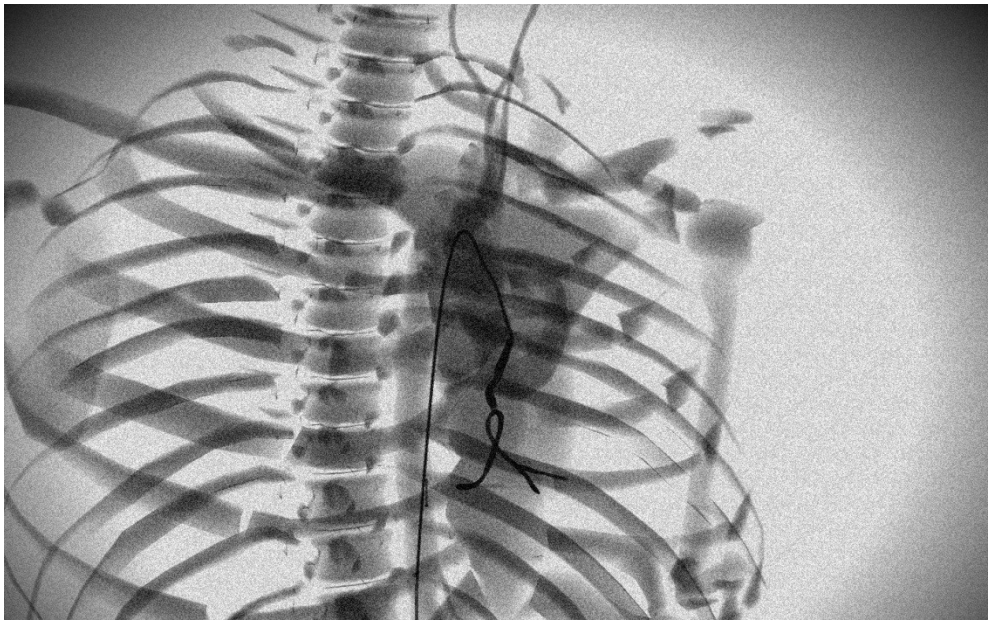


Figure 5.6: The simulated fluoroscopy screen of VCSim3 showing a catheter inserted into the RCA with contrast injected.

5.3.4.2 Cardiac motion

Several attempts were made to segment a detailed, animated heart mesh including the coronary arteries from a combination of CT and MRI images using computer vision methods. However, this approach did not yield the necessary results in the available time. Therefore, a hybrid approach was used instead. A static, coronary arteries mesh (Figure 5.7 left) was segmented from a real-patient data set and manually, frame by frame, overlaid onto the surface of an off-the-shelf animated (24 frames) heart model (www.3dscience.com, Figure 5.7 right) using a 3D software package (3ds Max, www.autodesk.com). The resulting sequence of 24 frames was then exported into a series of OBJ files readable by the simulator.

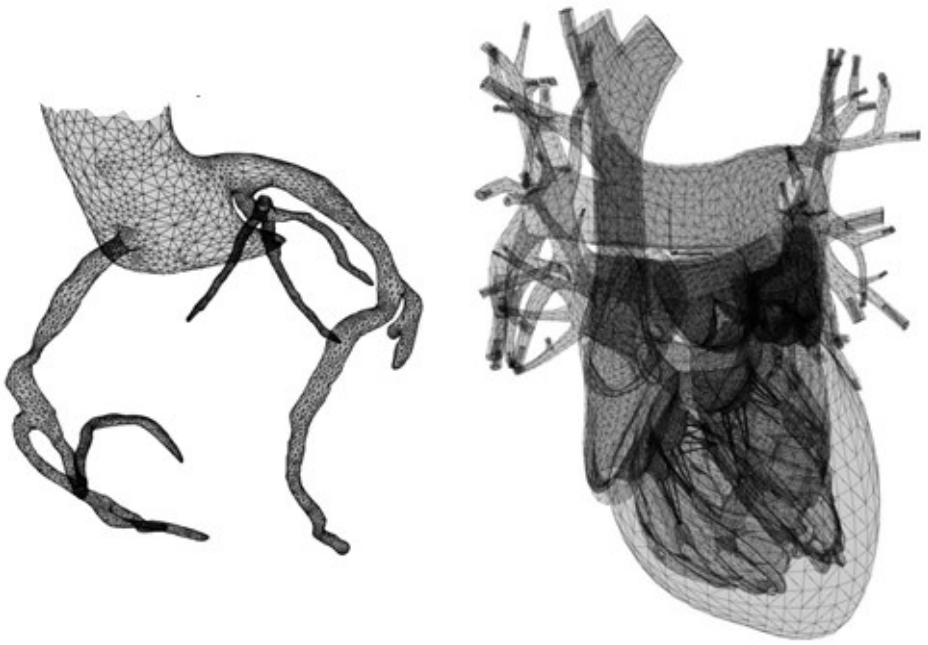


Figure 5.7: The wireframe rendering of coronaries extracted from real patient data (left) and off-the-shelf 3D heart model (right).

5.3.4.3 Contrast medium propagation

The contrast propagation model is a modified version of an approach by (Wang, 2009), which uses the vessel centreline to propagate the contrast agent. A contrast centreline can be thought of as an ordered set of spheres fitted into the coronaries. The spheres are grouped into branches. Each sphere is described by its index, position, radius and calculated flow speed. When contrast is being injected, the closest sphere is selected and the propagation starts. It is visualized by simply changing the opacity of the successive spheres according to its volume and amount of contrast currently flowing through it (Figure 5.8). The speed of this propagation depends on the blood flow speed, i.e. current heartbeat phase. Another mechanism is responsible for gradually fading away the contrast agent.

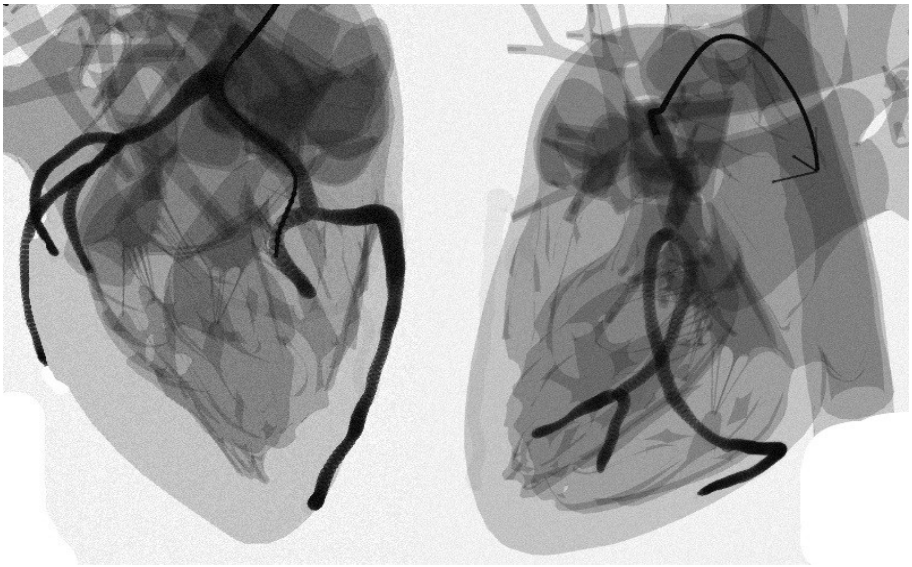


Figure 5.8: The virtual contrast medium under fluoroscopic visualization injected into right (RCA) and left (LCA) coronary artery.

The main difference to (Wang, 2009) is that the centrelines are no longer pre-generated in an automatic way from medical scans. Instead, a new, semi-automatic tool was developed, which uses the segmented sequence of coronary polygonal meshes. Therefore, it provides support for coronary motion. The user is responsible for indicating, in the editor, start, end and bifurcation points of the vessels in an initial frame. The editor then generates the intermediate spheres, fits them into the coronaries and calculates the flow speed using the major law of fluid dynamics – *the Poiseuille Law*. The obtained centreline is used to generate the successive frames. Figure 5.9 depicts a fitted centreline with manually set points coloured in red.

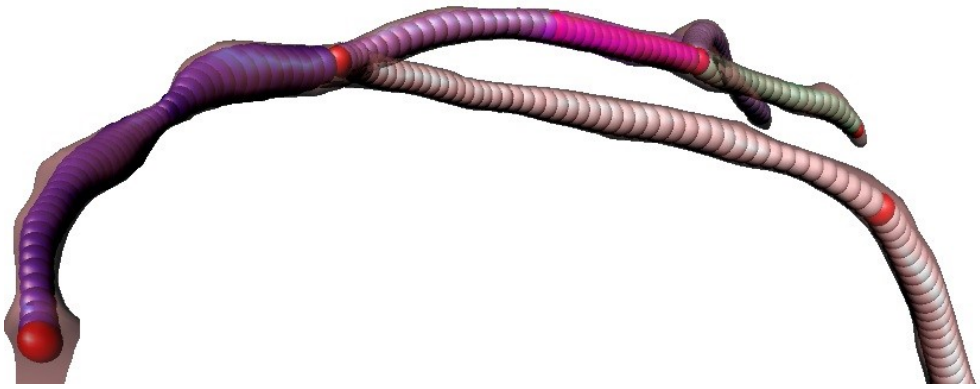


Figure 5.9: A visualized centreline fitted into the vessel.

5.3.4.4 Balloon inflation and stent deployment

The balloon is modelled as a number of spheres attached by stiff springs to the guidewire. During inflation, the radiuses of spheres are iteratively increased and dynamically adjusted to fit into the shape of the vessels. This is done by iterative collision detection and response of each sphere in order to calculate possible expansions and the best location inside the vessel. During inflation, the polygonal model of a stenosed vessel is gradually interpolated into the healthier one. The stent is modelled in a similar manner as the balloon. Until release, its mass-points follow the balloon. When released, its mass-points are separated and a mass spring model takes care of keeping it in place following the movement of the vessels. Figure 5.10 shows the main stages of stent deployment.

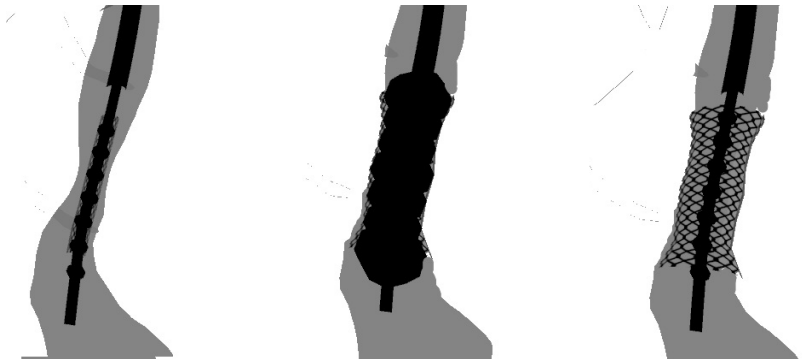


Figure 5.10: Three stages of stent deployment: Positioning the balloon with stent in a stenosis (left), inflating the balloon (middle), deflating the balloon and releasing the stent (right).

5.4 RESULTS

5.4.1 Parameters optimization

Table 5.1 summarizes the validated RMS error between the simulated instrument and the nearest corresponding points on the reference centreline (averaged) for each simulated instrument at the deepest insertion (left renal artery), as well as the average (using all three insertion depths) RMS error, Euclidean distance error, standard deviation and maximum Euclidean distance (mm). For two of the instruments (Rosen, Beacon), the accuracy obtained was at the sub-millimetre level, whilst for another two (Terumo ST4 and ST5 catheters) was slightly above. The highest error was for the stiff Amplatz guidewire - 4.33mm. Average result at the 3rd depth was 2.28mm.

Table 5.1: Comparison between real and simulated instruments

	3rd depth RMS (mm)	Avg. RMS (mm)	Avg. Eucl. Dist. (mm)	Avg. Std. dev. (mm)	Avg. Max. Dist. (mm)	Eb (1×10^7)	Radius (1×10^3)	Density (1×10^5)	Kp (1×10^3)
Amplatz	4.33	2.60	1.94	1.66	6.70	3.65	8.9	8.47	1.63
Cook Str.	2.84	1.78	1.31	1.15	5.01	3.28	6.5	3.09	39.9
Bentson	2.69	2.06	1.72	1.13	5.37	9.03	7.5	88.2	4.05
Terumo Stiff	2.90	2.02	1.59	1.20	4.99	5.45	6.2	15.0	31.2
Cook J	3.63	2.09	1.62	1.30	4.88	17.6	6.8	1.33	10.2
Rosen	0.91	0.81	0.63	0.51	4.04	3.90	6.7	7.24	37.7
Beacon 5FR	0.90	1.08	0.98	0.45	2.56	3.53	7.4	36.5	30.2
Terumo ST4F	1.18	1.33	1.19	0.58	3.57	3.66	6.2	1.80	8.92
Terumo ST5F	1.14	1.21	1.07	0.55	3.43	3.54	6.3	11.4	31.7
Average	*2.28*	*1.66*	*1.34*	*0.95*	*4.50*				

The average distance error between the simulated and scanned instruments, measured using all three insertions, was 1.34mm (standard deviation: 0.95mm, RMS: 1.66mm). Comparing this error to the corresponding error of 2.27mm presented in a previous study (Luboz et al., 2009a) with a mass-spring model using the same real instrument data set, there is nearly a 60% accuracy increase. The average RMS error (1.66mm) is slightly higher than the Cosserat implementation in (Tang et al., 2012) (1.25mm).

5.4.2 Compression

Chapter 3 section (3.2.2) presented a stretch of the rod suspended in a free space in respect to different number of solver iterations. Here, in Figure 5.11 and Figure 5.12, a real life scenario - a compression of the rod due to collision with a rigid anatomy is studied. The rod consisting of 256 Cosserat elements was gradually inserted into a silicone phantom model as to aggravate its compression. The distance constraint mixing factor (k_d) was set to 1.0 and the Baumgarte stabilization term (k_s) to 0.1. After each distance constraints iteration, the collision and friction (0.2) constraints were applied.

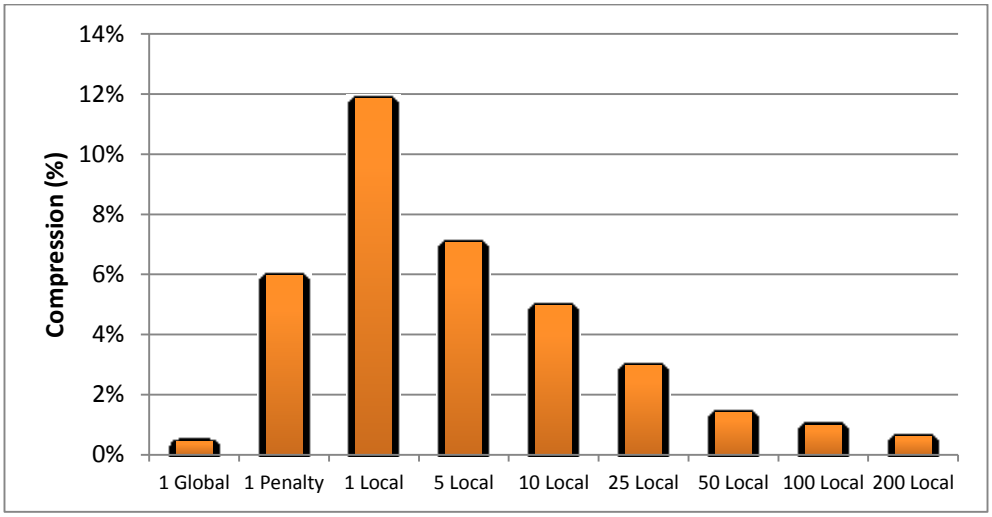


Figure 5.11: Rod compression in respect to different type and number of distance constraints iterations.

Similarly to the stretch experiment, a single global iteration was sufficient to keep the rod practically incompressible (0.45%). 25 – 50 (2.98% - 1.40%) iterations resulted in visually acceptable results as shown in Figure 5.12.

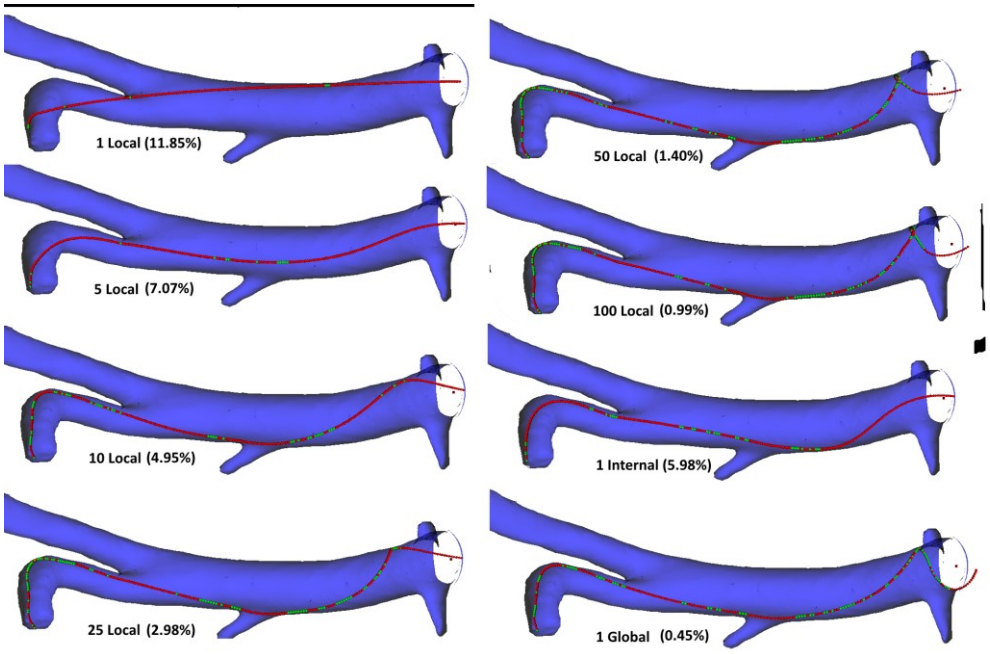


Figure 5.12: Rod compression in respect to different number and type of distance constraints iterations. The resulting percentage of compression is given in brackets. The green spheres represent colliding mass-points and the red ones - non-colliding.

Increasing the iterations beyond 50 did not improve the visual accuracy significantly. This agrees with the inverse exponential convergence curve of Figure 5.13, where compression reduction for 100 and 200 iterations is only 0.41% and 0.79% better, respectively, compared to 50 iterations.

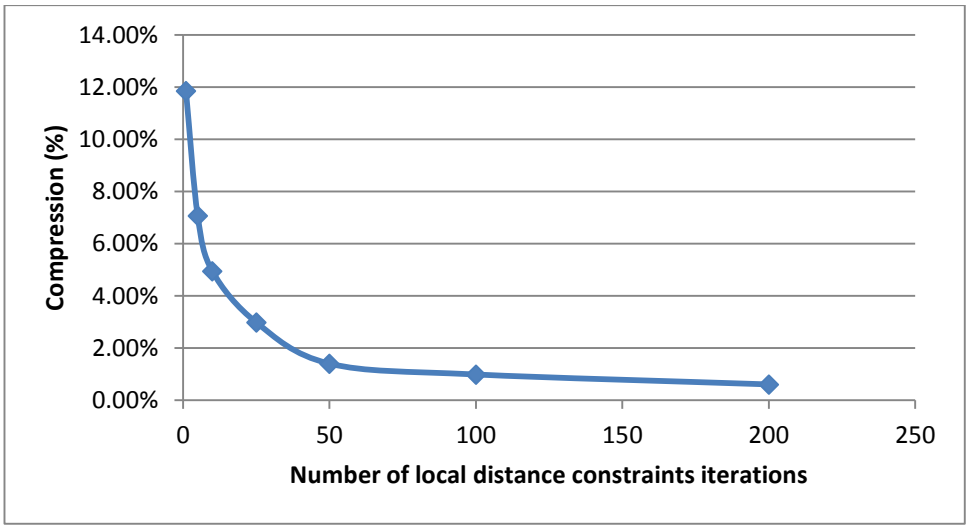
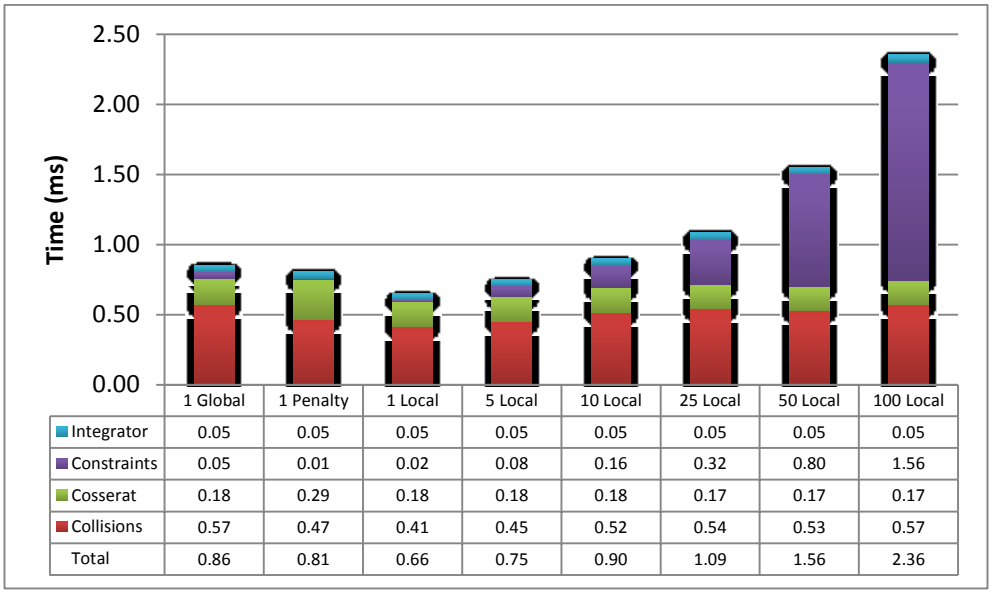


Figure 5.13: Percentage of compression in respect to the number of iterations

5.4.3 Computational Performance

The computational time measurements were taken on a HP x4600 workstation (Intel Core2 Quad @2.66 GHz, 8GB RAM) running Windows 7 x64. Figure 5.14 presents the computational time in respect to type and number of distance constraints iterations in the compression test, i.e. a single rod consisting of 256 Cosserat elements inserted into the silicone phantom model. Self-collisions were not considered.



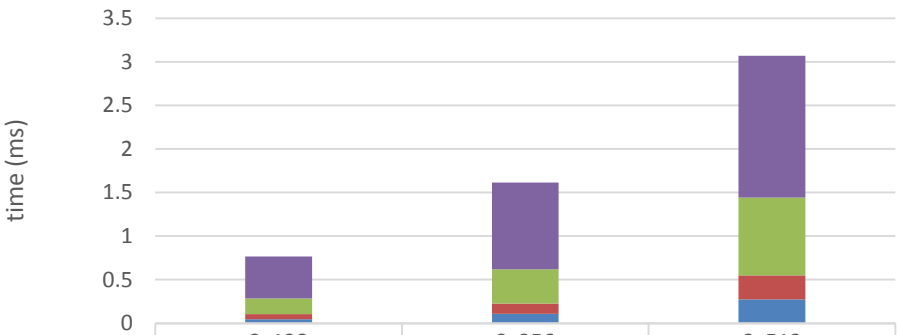
	1 Global	1 Penalty	1 Local	5 Local	10 Local	25 Local	50 Local	100 Local
Integrator	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05
Constraints	0.05	0.01	0.02	0.08	0.16	0.32	0.80	1.56
Cosserat	0.18	0.29	0.18	0.18	0.18	0.17	0.17	0.17
Collisions	0.57	0.47	0.41	0.45	0.52	0.54	0.53	0.57
Total	0.86	0.81	0.66	0.75	0.90	1.09	1.56	2.36

Figure 5.14: The computational performance of a single rod consisting of 256 Cosserat elements.

The total computation time of a single global iteration (0.85ms) is closest to 10 local iterations (0.90ms). Even though the global constraints calculations are 3 times faster (0.05ms vs. 0.16ms), an increased number of collisions caused by the lower rod compression reduces the performance (0.57ms vs 0.52ms). The local constraints calculation time increases linearly in respect to the number of iterations. For both 1 global and 10 local iterations, the collision detection is a dominant part of the calculations, taking 67% and 58% of the total calculation time, respectively, which causes a serious performance bottleneck. Fortunately, the collision detection may be separated into a different thread (task parallelism, see Section 4.3.6 for details).

5.4.4 Catheter/guidewire pair performance

Figure 5.15 shows the computational performance of a catheter/guidewire pair inserted into the aforementioned silicone model. The experiment was run on the same test platform as the previous one (Intel Core2 Quad @2.66 GHz, 8GB RAM). The guidewire was inserted into the model first, followed by the catheter. The instruments interacted both ways i.e. the dynamics of the catheter were affecting the guidewire and vice versa. Self-collisions were not considered.



	2x128	2x256	2x512
Collisions	0.48	0.997	1.625
Cosserat	0.18	0.392	0.893
Constraints	0.057	0.117	0.275
Integration	0.048	0.109	0.275
Total	0.765	1.615	3.068

Figure 5.15: Averaged computational times in milliseconds including instruments interactions achieved during insertion of guidewire/catheter pair into a silicone phantom model for different rod lengths.

From the above Figure 5.15, it can be seen that the binding constraints introduced a minimal performance overhead. Similarly to the previous test, the collision detection took most of the computational time. In this case, the rods consisting of 512 Cosserat elements would not a run at minimum haptic interactive rates (500Hz, <2ms per physics step) required for smooth force feedback. However, as mentioned before, it could be allocated to a different thread (see Section 4.3.6). For the 2x512 case, this would result in a total physics computation time of 1.44ms, which would still meet the haptic interactivity requirement.

5.4.5 Angioplasty procedure

The following figures depict the stages of the simulated angioplasty (stenting) procedure. During the procedure, an interventional cardiologist inserts the catheter/guidewire pair into the femoral artery. By pushing, pulling and rotating the proximal end of the tool pair, s/he navigates the instruments into the heart coronaries using real time fluoroscopy imaging. Figure 5.16 shows the steps of the instruments reaching the aortic arch (1). In (2), the instruments are inserted (or "dropped") into the heart, are rotated to point into the RCA (3) and pulled back until they "jump" into the coronary entry.

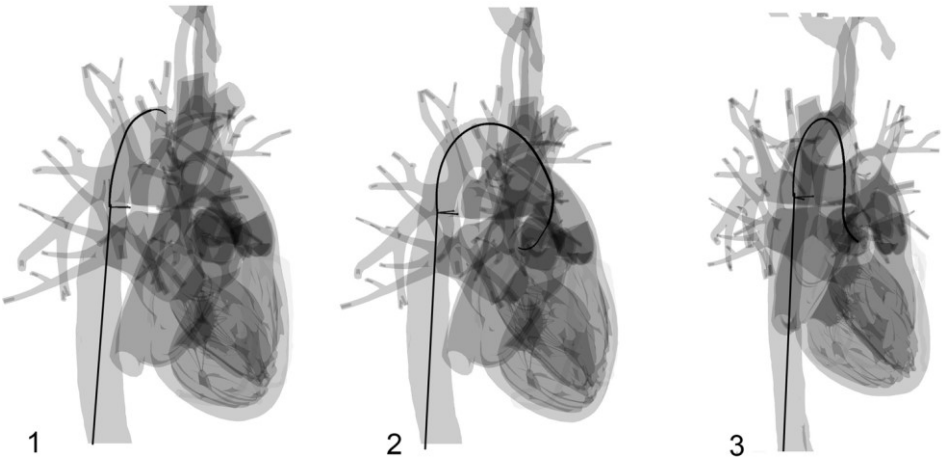


Figure 5.16: Navigation of instruments into the RCA: (1) Instruments reach the aortic arch. (2) Catheter and guidewire bends and slide into the heart. (3) Catheter is pointed into the RCA direction.

Figure 5.17 shows the next steps of the procedure. When the catheter "sits" inside the coronary entry (4), the contrast is injected in order to locate the narrowing (5). Next, the guidewire with the stent and balloon is introduced into the RCA (6) at the height of the stenosis.

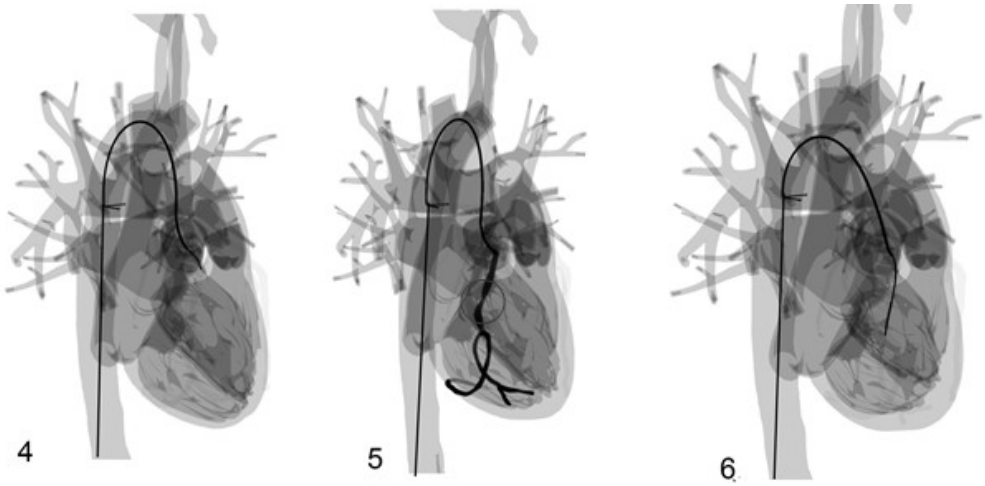


Figure 5.17: Locating the narrowing and balloon positioning: (4) Guidewire slides into RCA, the catheter follows. (5) Guidewire is removed and contrast is injected. (6) The balloon is inserted at the height of the narrowing.

Figure 5.18 depicts the last three stages of the procedure. A balloon with a stent is inflated expanding the narrowing (7). After the balloon is deflated, the stent stays in place, preventing the vessels from collapsing (8). Finally, the balloon is removed and the contrast is injected to evaluate the outcome. The procedure is complete and the instruments can be removed.

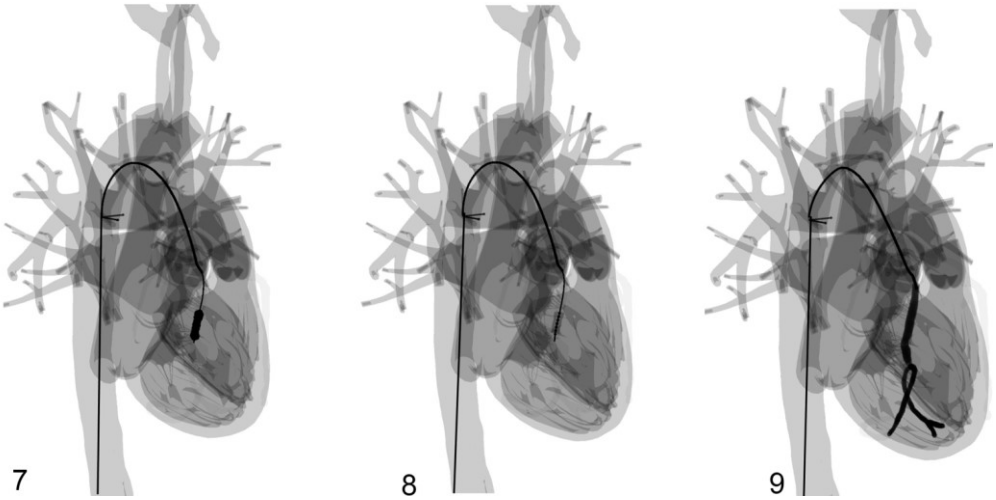


Figure 5.18: Balloon inflation and stent release: (7) Balloon with self-expanding stent is inflated expanding the vasculature. (8) Balloon is deflated; the stent is fully expanded and stays in place. (9) Contrast is injected to see the final outcome.

5.4.6 Validation

An experimental study was designed in order to assess the face and content validity of the VCSim3 simulator. The participants were required to complete 5 cardiovascular interventions. Specifically, to navigate the catheter and guidewire from the femoral artery into the heart coronaries, localize the stenosis and deploy a stent. Prior to performing their first procedure, all participants were given a technical instruction sheet outlining the nature of the simulation (see Appendix D). The aim of this sheet was to give a brief overview of the equipment, simulator software, functionality and task to perform. After reading the instruction sheet, participants were given a maximum of 2 minutes to familiarize themselves with the manipulation of instruments. Prior to commencing their first attempt, participants were given the opportunity to ask questions relating to the practicalities of the simulation, but were not allowed to request any technical advice as to how best to perform the procedure. The study was given a favourable ethical opinion for conduct by the Imperial College Research Ethics Committee (ref. ICREC_14_2_9).

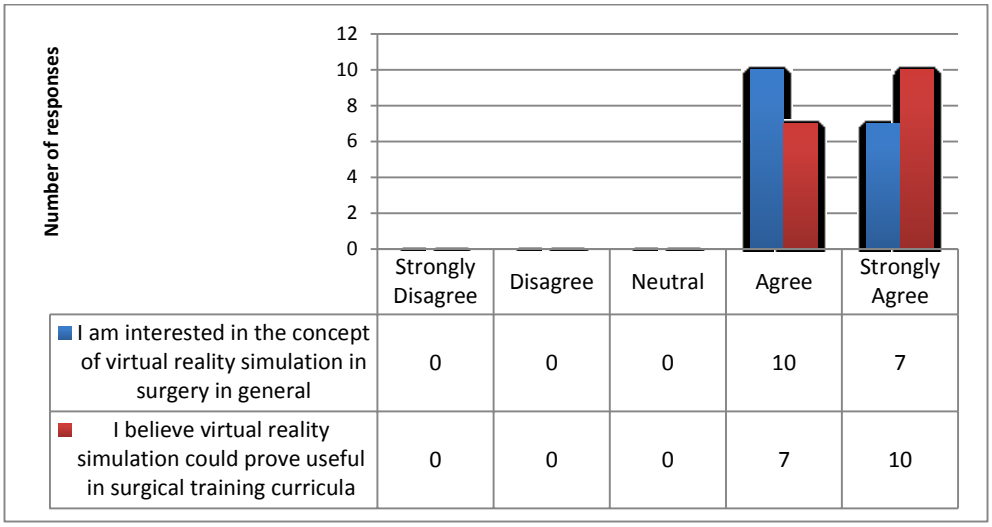
5.4.6.1 Demographics

17 participants (15 males and 2 females) were recruited for the study. All of them described their medical specialization as “cardiology”. Table 5.2 summarizes the previous experience of the participants.

	Average	Median	Min	Max
Postgraduate year of training (PGY)*	7.12	7	2	13
Procedures in humans	1166.18	600	5	5000
Procedures on VR simulators	6.94	2	0	50
Procedures on physical simulators	19.82	10	0	150

Table 5.2: Participants PGY and experience

Figure 5.19 shows that the interest in the concept of virtual reality in surgery in general, and their belief that it could prove useful in the surgical training curriculum was high. All the participants responded that they either “Agree” or “Strongly Agree” with these statements.



	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
I am interested in the concept of virtual reality simulation in surgery in general	0	0	0	10	7
I believe virtual reality simulation could prove useful in surgical training curricula	0	0	0	7	10

Figure 5.19: Interest in concept of VR simulation

5.4.6.2 Face validation – instruments behaviour

Face validity (Figure 5.20) was evaluated by asking participants to complete a questionnaire (Appendix D) after completing the study. 11 questions related to the behaviour of the virtual instruments and its interactions with the vessels. The specific questions were:

Q1: The lengths of catheter/guidewire were realistic

Q2: The catheter/guidewire bending behaviour was realistic

Q3: The catheter/guidewire twisting behaviour was realistic

Q4: The catheter/guidewire (non-)stretching behaviour was realistic

Q5: The catheter/guidewire tip behaviour was realistic

Q6: The catheter/guidewire body (shaft) behaviour was realistic

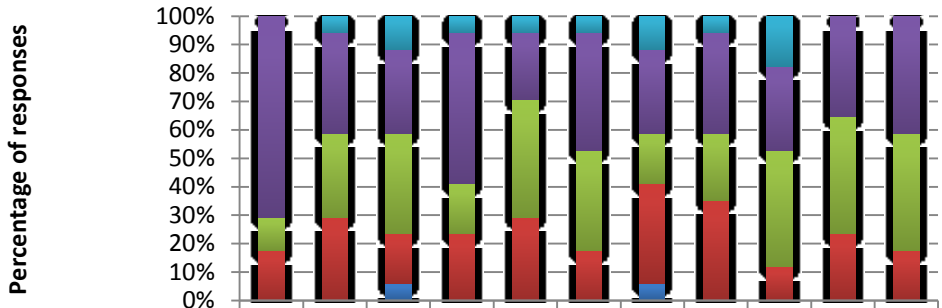
Q7: The interactions between catheter and guidewire were realistic

Q8: The interactions between catheter/guidewire and heart vessels were realistic

Q9: The delay between physical manipulation and visual reaction was realistic

Q10: The haptic force feedback felt realistic

Q11: Overall, the catheter/guidewire behaviour felt realistic



	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Q11
Strongly Agree	0	1	2	1	1	1	2	1	3	0	0
Agree	12	6	5	9	4	7	5	6	5	6	7
Neutral	2	5	6	3	7	6	3	4	7	7	7
Disagree	3	5	3	4	5	3	6	6	2	4	3
Strongly Disagree	0	0	1	0	0	0	1	0	0	0	0

Figure 5.20: VCSim3 face validity of instruments behaviour - Likert scale responses

70% of participants agreed that the lengths of instruments were realistic (Q1). In terms of questions assessing the physical behaviour of the instruments (Q2-Q6), 43% of responses were “agree” or “strongly agree”, 32% were neutral, 24% “disagree” and 1% “strongly disagree”. In terms of questions assessing interactions between instruments (Q7) and vessels (Q8), 41% of responses were “agree” or “strongly agree”, 21% were neutral, 35% “disagree and 3% “strongly disagree”. Nearly half (47%) of respondents were positive or very positive about the latency (Q9) and 35% were affirmative about the haptic force feedback (Q10). In both statements (Q9 and Q10), 41% of participants were neutral with 12% and 24% disagreeing.

41% of participants agreed that, overall, the catheter/guidewire behaviour was realistic (Q11) with 41% being neutral and 18% disagreeing. In total 45% of the responses to the questions assessing the realism of the behaviour of the instruments (Q1-Q11) were “agree” or “strongly agree”, 30% were “neutral”, 25% “disagree” and 1% “strongly disagree”.

The participants had an opportunity to leave free text comments:

- *"The overall catheter/guidewire interaction was quite realistic. However, the manipulation of the catheters was not so realistic and they didn't feel like real catheters as they moved too smoothly."*
- *"The guidewire torque was not realistic as it was very heavy compared to normal."*
- *"The wire popped out of the catheter unexpectedly on a frequent basis in a way it wouldn't normally do. The appearance of the wire was also not accurate as it appeared to be beaded."*
- *"Manipulation of the catheters was easier than in real life."*
- *"The latency was probably a little too short (i.e. too responsive) and the catheters felt a little "floppy"."*
- *"When tried to withdraw the catheter back while fixing the guide wire, the guide wire moved distally in the coronary vessel."*
- *"Great concept, but still not quite realistic feeling."*

5.4.6.3 Face validation – supporting solutions

Eight questions were asked concerning the realism of the supporting solutions (Figure 5.21). The specific questions were:

Q1: The contrast medium injection felt realistic

Q2: The balloon inflation felt realistic

Q3: The contrast medium propagation was visually realistic

Q4: The balloon inflation was visually realistic

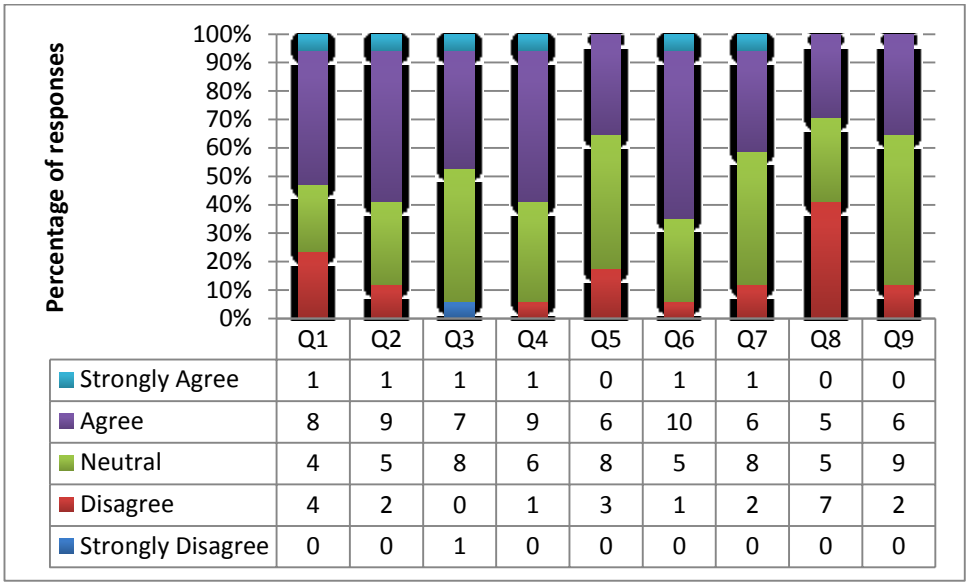
Q5: The stent deployment was visually realistic

Q6: The cardiac motion was visually realistic

Q7: The visualization as a whole looked realistic

Q8: The difficulty of the simulated procedure was realistic

Q9: Overall the simulator was realistic



	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9
Strongly Agree	1	1	1	1	0	1	1	0	0
Agree	8	9	7	9	6	10	6	5	6
Neutral	4	5	8	6	8	5	8	5	9
Disagree	4	2	0	1	3	1	2	7	2
Strongly Disagree	0	0	1	0	0	0	0	0	0

Figure 5.21: VCSim3 face validity of supporting solutions - Likert scale responses

In terms of supporting solutions, 56% of answers to statements regarding the feel of the contrast medium injection and balloon inflations (Q1-Q2) were “agree” and “strongly agree”, 26% were “neutral” and the remaining 18 were “disagree”. 50% of responses evaluating the visual aspects of the simulator (Q3-Q7) were “agree” and “strongly agree” and 41% were “neutral”. 41% of participants disagreed that the difficulty of the simulated procedure was realistic (Q8), 29.5% were “neutral” and 29.5% “agreed”. 35% of participants agreed that, overall, the simulator was realistic (Q9) with 53% being neutral and 12% disagreeing with this statement.

In total, 47% of all the responses to questions (Q1-Q9) assessing the realism of the supporting solutions were “agree” or “strongly agree”, 38% were “neutral” and 14% “disagree”. Only one response was “strongly disagree”

Free text responses included:

- *“It's a little too responsive and easy. This is true of all the simulators I've used. The visualisations were, if anything, too good (i.e. the definition was sharp and the anatomy too easily seen). In the cath lab, we can't see the vessels we're aiming for until we find them.”*

- *"Manipulating the guidewire was too difficult. The balloon manipulation and inflation was very realistic."*
- *"Needs improved tracking/force feedback equipment. The behaviour of the tip of the catheter was very impressive - jumping up, storing torque etc. into the coronary arteries. Also good was how it would occasionally jump up the carotids mirroring cath lab problems. Liked the push back on the catheter that you got when you post the intra-coronary wire down the coronary artery with it sometimes tangling up the shape of the catheter and pulling out the wire - also mirrored the tricks you do to restore the correct positioning."*

5.4.6.4 Content validation

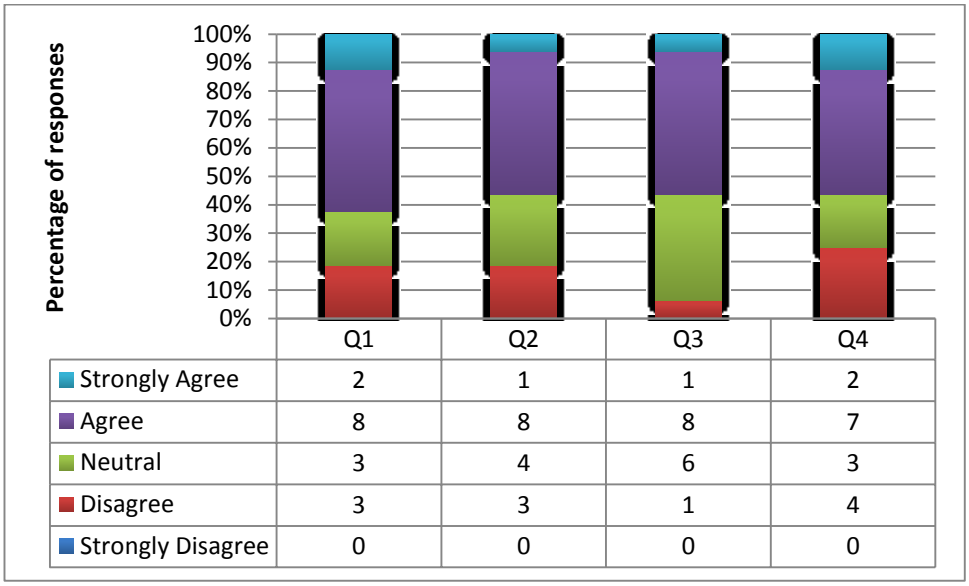
Content validity was also evaluated. The questions asked assessed the adequacy of the simulated tasks and perceived utility of the simulator as a training tool for cardiovascular interventions (Figure 5.22). Specifically, it contained 4 questions:

Q1: The catheter/guidewire behaviour is sufficient to make it a useful training tool for cardiovascular interventions.

Q2: The remaining functionality provided by the simulator is sufficient to make it a useful training tool for cardiovascular interventions.

Q3: Overall the simulator is a useful training tool for cardiovascular interventions

Q4: I would recommend the simulator to others.



	Q1	Q2	Q3	Q4
Strongly Agree	2	1	1	2
Agree	8	8	8	7
Neutral	3	4	6	3
Disagree	3	3	1	4
Strongly Disagree	0	0	0	0

Figure 5.22: VCSim3 content validity - Likert scale responses

In total, 58% of the responses to the questions assessing the realism of the supporting solutions were "agree" or "strongly agree", 25% were "neutral" and 17% "disagree". None of the participants "strongly disagreed" with any of the above statements.

Free text responses included:

- *"At a very early stage in training there is probably a role for this in demonstrating the coronary anatomy in relation to different radiological views. Basic intubation concepts and procedural concepts could also be usefully practised."*
- *"Tactile feedback needs more work."*
- *"This has huge potential, but trainees could not yet rely on this simulator as a reflection of what happens in patients."*
- *"So far the simulator is not very realistic and needs a lot of work still. The manipulation of wires is the main problem."*
- *"Clearly has great potential and the core behaviours are done very well. Needs a few tweaks to the physical interface, and reflect some of the*

constraints you have in the cath lab - only a few fixed views, less clear images etc."

5.4.6.5 Discussion

The results of the face validity suggest that the VCSim3 demonstrates early signs of face validity in terms of the realism of the instruments. However, given the free text comments from participants, it is likely that the drawbacks of the VSP haptic device affected their perception of the realism of the virtual instruments. The limited catheter tracking length (30cm) is, admittedly, sufficient to navigate the catheter between the aorta and the coronaries but, having to insert the catheter too deep in order to reach the second sensor, "hides" the guidewire, resulting in a loss of tracking of the wire, which in turns makes the wire unresponsive. This requires retracting both tools out of the VSP and reinserting them, which was frustrating and confusing for the participants.

Moreover, in order to guarantee more accurate optical instrument tracking, particularly of the guidewire, a physical instrument with a non-uniform surface was preferred. Using smooth and glossy wires resulted in frequent skips in tracking and occasionally complete "loss" of the wire. To address this, a coated wire was used, but this was stiffer than the virtual instrument. This addressed the tracking problem, but might have contributed to the confusion of the participants since the wire they saw on the screen behaved slightly different than the one they manipulated with their hands.

Finally, the force feedback mechanism inside the VSP is built around a motor which simply clamps the instruments. Such a mechanism does not separate linear force feedback from rotational force feedback. These disadvantages can be potentially solved by using an improved haptic device.

The results of the face validity of supporting solutions, such as fluoroscopic visualization, contrast propagation, cardiac motion, balloon inflation and stent deployment, range mainly from neutral to positive. This suggests that, although

technically simple, they provide sufficient functionality and realism in the view of the majority of the participants.

In statements relating to content validity, more than 50% of responses were positive or very positive indicating that VCSim3 has the potential to become a useful training tool for cardiovascular interventions. Most of the participants would also recommend it to their colleagues.

5.5 SUMMARY

This chapter introduces VCSim3 – a VR simulator for cardiovascular interventions. The behaviour of the virtual catheter and guidewire is based on the inextensible Cosserat rod implementation presented in Chapter 3. The model allows for efficient modelling of bending, stretching and twisting phenomena, as well as guaranteeing almost immediate response to user manipulations, even for long instruments. The mechanical parameters of six guidewires and three catheters were optimised with respect to their real counterparts scanned in a silicone phantom using CT. The implementation allows the simulator to run efficiently on an off-the-shelf PC or laptop, significantly exceeding the minimum required haptic interactive rate.

Results show the parameter-optimised virtual instruments exhibit near sub-millimetre accuracy, with errors likely to be caused by the accidental rotations and resulting torsion introduced during the insertion of real instruments into the silicone phantom. However, due to the availability of only one dataset, the parameter optimization at two insertion points and verification at the 3rd one could potentially result in overfitting.

The global distance constraints used result in a practically inextensible and incompressible rod. The compression of the rod consisting of 256 Cosserat elements, which was inserted into a silicone phantom model in a way as to aggravate this effect, was below 0.5%.

An initial verification of the simulator was carried out by obtaining subjective feedback (face and content validity) from 17 cardiologists. The results of the face validity suggest that VCSim3 demonstrates its early signs in terms of the realism of the simulated instruments. Nearly half of the participants was positive about the behaviour of virtual catheters and guidewires. The results of the face validity of supporting solutions such as fluoroscopic visualization, cardiac motion, contrast propagation, balloon inflation and stent deployment ranged from neutral to positive. In terms of content validity, more than half of responses were positive or very positive indicating that the majority of the participants agreed that VCSim3 is

a useful training tool for endovascular interventions, and that they would recommend it to others.

In terms of limitations, VCSim3 suffered from the restrictions of the haptic device mentioned earlier. The number of available procedures is also currently limited to angioplasty and stenting of left and right coronary arteries extracted from a single patient-specific CT dataset. The catheter and guidewire behaviour, although on average positively acknowledged by the participants, received more mixed feedback than the virtual endoscope in NOViSE. This could be due to the more subtle nature of the guidewires and catheter manipulation, as well as the shortcomings of the haptic device. Therefore, more investigation is needed in order to identify some subtle phenomena occurring during the real endovascular procedures.

The supporting solutions implemented in the simulator, although visually plausible and positively rated by the participants, lag behind the state-of-the-art. The face and content validation study suffered from relatively small number of participants and the construct validity was not assessed. Therefore, it is intended to repeat the study with a larger number of participants and including construct validation after eliminating the above-mentioned problems.

Performance-wise, the inextensible Cosserat Rod is efficient and able to deliver real-time haptic experience for reasonably long rods on a modern consumer level computer. However, in order to achieve satisfactory results for longer virtual instruments extending from the femoral artery, some accuracy would need to be sacrificed. To address this issue, the next chapter focuses on accelerating the virtual instruments through a massively parallel implementation running on graphics processing units.

Chapter 6

GPU ACCELERATION

In the previous chapter, the Cosserat Rod model presented in Chapter 3 was used to simulate a catheter/guidewire pair in a cardiovascular procedure. Although the model was computationally faster than similar solutions, particularly when considering its inextensibility and incompressibility, in the case of the simulation of very long instruments, it was still not fast enough to deliver real-time performance at haptic interactive rates on the target platforms. One of the methods constantly gaining acclaim in the physically-based simulation field, is to massively-parallelize the computations using graphical processing units (GPUs). GPUs are considered as processors that can deliver a high performance boost (1-2 orders of magnitude) for some type of algorithms, especially when a problem set is large (at least tens of thousands elements). As such, an elastic rod may not seem as a good candidate for this method. Indeed, such attempts found in existing literature have not resulted in substantial speed gains.

In this chapter, a massively-parallel GPU implementation of the CoRdE model and its modified inextensible version is presented. By superseding the *CUDA Scalable Programming Model* and using inter-block synchronization, it is possible to simulate multiple physics time-steps per single kernel launch utilizing all the GPU streaming multiprocessors.

6.1 GPGPU ON NVIDIA CUDA

In recent years, graphics processing units (GPUs) have evolved into highly parallel multi-core systems allowing solving of general-purpose problems (GPGPU computing). Unlike CPUs, GPUs have an architecture oriented on throughput, specialized in computationally-intensive, highly parallel computation, rather than complex data caching or flow control (Figure 6.1). Because the same program is executed for each data element, there is a lower requirement for sophisticated flow control. In addition, since the same program is executed on many data elements and has high arithmetic intensity (the ratio of arithmetic operations to memory operations), latencies due to memory access can be hidden without the need of big data caches.

Figure 6.1: CPU vs GPU architecture comparison. (NVidia, 2014)

CUDA (Compute Unified Device Architecture) is a general purpose parallel computing platform and programming model implemented for the NVidia GPUs (NVidia, 2014). It gives developers access to the virtual instruction set and memory of the parallel computational elements in CUDA GPUs. A CUDA enabled GPU is built around a set of Streaming Multiprocessors (SMs, Figure 6.2). Each SM consists of a number of compute cores (CUDA cores) depending on the chip generation (8-192). Each CUDA core contains an array of integer and floating point arithmetic logic units (ALUs) and special function units (SFUs), which handle specialized instructions such as trigonometric functions, square roots or reciprocals.

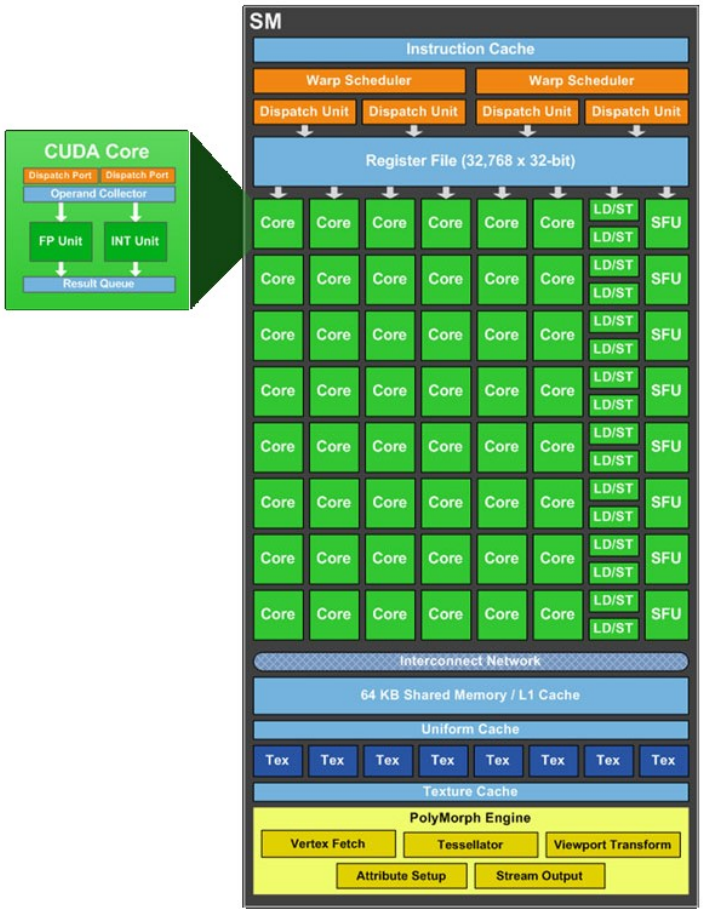


Figure 6.2: Streaming multiprocessor and CUDA core of Fermi architecture. (NVidia, 2014)

Each SM is also equipped with a relatively small amount (48-64KB) of fast on-chip memory called shared memory (Figure 6.3). Shared memory allows threads within the same block to cooperate on solving a sub-problem. It also enables reuse of data and reduces the traffic to / from off-chip global memory (DRAM, 1-4GB). For many CUDA applications, exploiting the shared memory is the key to achieving high performance gains.

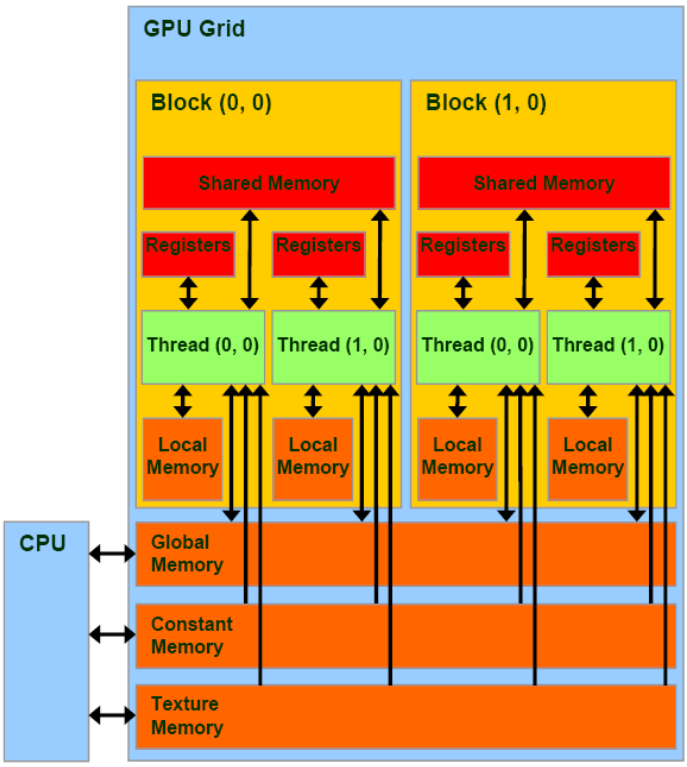


Figure 6.3: CUDA memory model. (NVidia, 2014)

A multithreaded CUDA program – a kernel – is partitioned into blocks of threads that execute independently from each other. The threads of a block execute concurrently on one SM, and multiple blocks (up to 8 on current generation) can execute concurrently on one SM. As blocks terminate, new blocks are launched on the vacated multiprocessors. In fact, each block can be scheduled on any of the available SM within a GPU, in any order, concurrently or sequentially, so that a CUDA program can execute on any number of multiprocessors located even on different physical GPUs or computing nodes. This decomposition is called *Scalable Programming Model* and is illustrated in Figure 6.4.

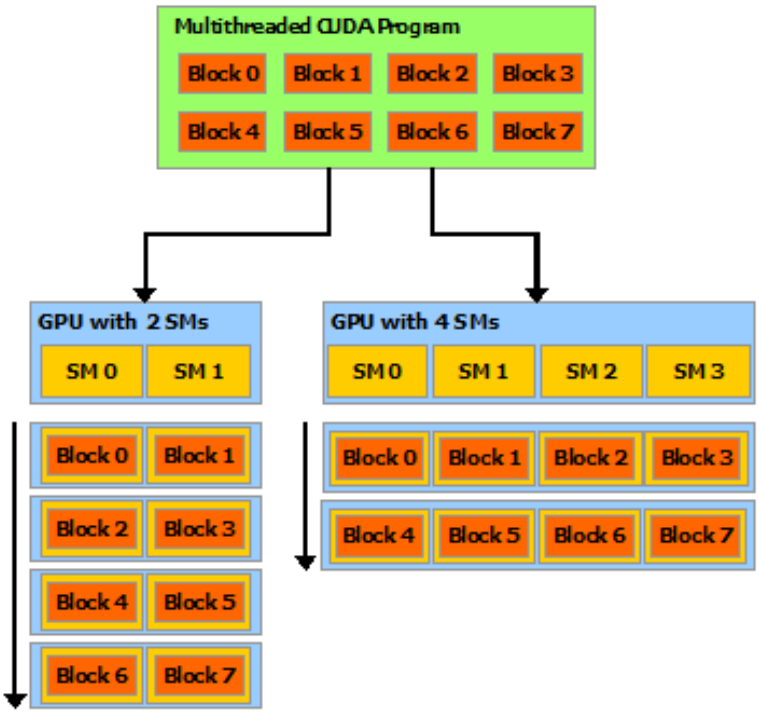


Figure 6.4: Scalable Programming Model of CUDA architecture. (NVidia, 2014)

It allows threads to cooperate within the thread block when solving each sub-problem, and at the same time enables automatic scalability. The *Scalable Programming Model* allows for synchronization of all the threads in the same block (intra-block sync) by *__syncthreads()* function calls. The CUDA specification does not define a function for a global, GPU-wide, inter-block synchronization as this would not allow the flexibility in blocks scheduling and, as a result, affect the overall scalability. The only way of doing this supported by the CUDA specification is to finish the execution of one kernel, and then launch a new one. However, by accepting some trade-offs, it is possible to supersede the *Scalable Programming Model* and implement inter-block synchronisation. A range of different methods for implementing such synchronisation is presented in (Shucai and Wu-chun, 2010).

6.2 RELATED WORK

An attempt to accelerate their simplified CoRdE implementation for suture simulation using CUDA was made by Punak et al. (Punak and Kurenov, 2011a), but the speed-up achieved was unsatisfactory and they decided to develop their model on the CPU in a serial manner. In (Kmoch et al., 2010), the authors applied CUDA to their hair animation system based on (Bergou et al., 2008), but did not report substantial speed gains either. In (Courtecuisse and Allard, 2009), Courtecuisse and Allard studied several parallel methods for solving dense linear systems using the Gauss-Seidel method. By carefully handling and overlapping computations, they reported speed-ups for larger problems (10000×10000) of an order of magnitude compared to a sequential CPU implementation. They applied their method to a medical intervention simulator where it was used to solve a linear complementary problem (LCP) for solving contact constraints of virtual catheters and coils in vessels. However, having other simulation components such as collision detection and construction of the mechanical compliance matrix running on the CPU, the overall simulation speed-up using the GPU was only 2-3x.

6.3 METHODS

The multi-threaded implementation consists of three main threads: physics simulation thread (Cosserat rod, collision detection), graphics thread and input thread (haptics).

The physics simulation can be toggled between the CPU and the GPU. During each GPU update, a single CUDA kernel is invoked taking as parameters the number of time-steps to process, the number of iterations over the constraints, material properties and input from the user (haptic device position vector). The graphics thread runs at standard 60Hz in the CPU. During each update, it requests a copy of the Cosserat rod mass-points positions to be transferred from the GPU to CPU for rendering purposes.

The thread handling communication with the input (haptic) device runs at 0.5-1 kHz and can request the state of the force-feedback vector, as well as provide a new position vector at any time from the physics simulation thread. This is possible because CUDA GPUs are equipped with separate data copy engines independent of kernel execution. This allows for fast, synchronous or asynchronous transfers (using pinned memory and different CUDA stream) of the intermediate data between the global memories of the GPU and CPU without waiting for the kernel to finish processing the specified number of physics simulation iterations. This enables writing the current state of the haptic device to the running kernel and getting back the force-feedback value at the rate higher than that of actual kernel executions.

6.3.1 Simple Parallelization

All the data arrays containing the initial positions of mass-points and quaternions, velocities and external forces are copied to the global memory of the GPU at the application initialization. During each kernel launch, i.e. physics simulation update, these arrays are loaded from the global memory (DRAM) to the fast shared memory of the SM in order not to duplicate costly global memory operations. Each thread processes a single Cosserat element, i.e. a mass-point and a corresponding quaternion. To achieve this, it requires access to information about the position

and velocity of the next neighbouring element. Processing a single time-step per kernel launch (1ts/kl) on the GPU can already yield a performance boost. However, the available arithmetic power is largely counter-balanced by the overhead of launching a kernel and global memory access latencies (Figure 6.5).

6.3.2 Multiple time-steps per kernel launch in a single block

Larger speed-ups are achieved by running multiple time-steps per kernel launch (*n* ts/kl, Figure 6.5). In a single-block kernel implementation, a specific case when all the mentioned arrays fit into the shared memory of a single SM can be done within the boundaries of the *Scalable Programming Model*. By careful intra-block synchronization between the threads using the *__syncthreads()* CUDA keyword, it is possible to keep the data in the shared memory between the subsequent time-step iterations. Additionally, calling *__syncthreads()* ensures shared memory consistency, i.e. changes made by one thread are visible to all the other threads within the same block. Running multiple iterations over the constraints requires additional position and velocity information exchange between the neighbouring mass-points, and thus two extra *__syncthreads()* calls per constraints iteration.

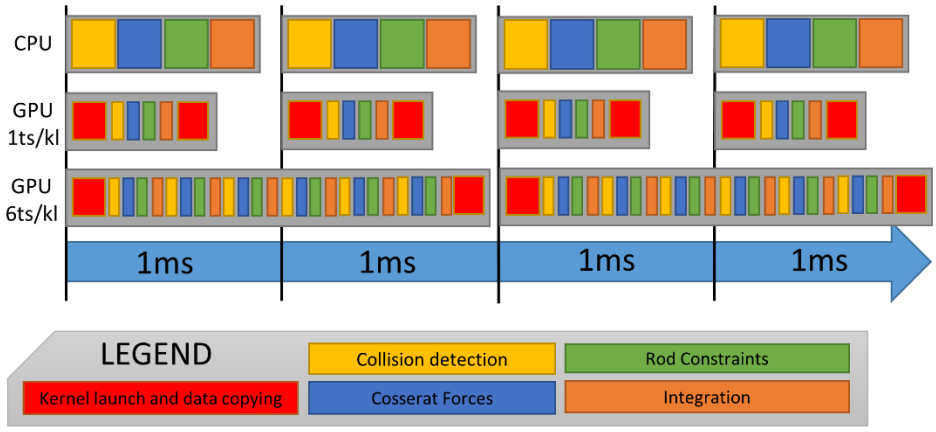


Figure 6.5: The concept of multiple physics time-steps per single kernel invocation

In this single-block case, small (48-64KB per block) shared memory size limits the number of the Cosserat elements which may be processed to around 512, which in turn limits the maximum length or accuracy of the simulated rod. As the current

generation of GPUs can handle many more threads per block (currently up to 2048), it may be possible to fit more Cosserat elements into the block by sacrificing some performance and moving data back to the global memory. However, considering handling collision detection during each time-step, which requires each thread to maintain its own stack in the shared memory for efficient iterative AABB tree traversal, the implementation in a single block is able to handle only around 256 mass-points maximum.

Utilizing just a single block occupying a single SM on a GPU that currently can have up to 15 of them (e.g. NVidia GeForce GTX780Ti) is clearly not an efficient use of resources. As mentioned earlier, the *Scalable Programming Model* does not allow for global, inter-block synchronization and communication. In the case of a multi-block kernel, such functionality is essential to correctly simulate even a single time-step per kernel launch with Cosserat elements spanned along several SMs. For an extensible rod, when using penalty forces rather than iterative distance constraints, this problem may be solved by splitting the kernel into two smaller kernels as the only possible way of GPU-wide syncing is by launching a new kernel. The first kernel would handle only the calculations of elastic forces. The second one would handle the integration of forces and velocities. This obviously adds launch overhead and, more importantly, prevents keeping the data in the shared memory as it is discarded at each kernel completion.

However, by superseding the *Scalable Programming Model* and accepting some trade-offs in portability across platforms / GPUs, it is possible to utilize all of the SMs available on the GPU. Using an atomic counter held in the global memory, a *__syncgpu()* method, which works as a barrier blocking the complete thread block until other blocks reach the same point (Shucai and Wu-chun, 2010), may be implemented. In this case it is not possible to schedule more blocks than there are SMs on the GPU, as this would result in some blocks never being processed, and others waiting indefinitely for them to reach the same point. The GPU would stall often resulting in a system crash.

Another problem is the inter-block or rather inter-SM communication. Data held in shared memory is only visible by threads residing in the same block. The only way to communicate between the blocks running on different SMs is via global memory (Figure 6.3). Given that in the Cosserat rod simulation there is mainly a need to exchange the data between the last and the first mass-point/quaternion of the bordering blocks, the performance penalty is not that substantial. To make this work, the pointers referring to global memory data must be marked as *volatile.* This ensures that all the memory accesses will result in an actual memory read/write operation, i.e. data will not be cached in registers or L1 cache.

```
load the data from the global to shared memory
for i physics iterations (time-steps)
{
        detect collisions
        apply external forces and torques
        calculate Cosserat elastic forces and torques
        integrate forces and torques
        for j constraint iterations
        {
                apply distance constraints impulses
                apply collision response impulse
                if(selfCollisions)
                        apply self-collisions impulse
                if(rodsBinding)
                        apply binding impulses
        }
        integrate linear and angular velocities
}
write the data from shared to global memory
```

Figure 6.6: The algorithm of the multi-block, inextensible Cosserat rod kernel. The lines in italic indicate the functions which require data exchange with other Cosserat element(s) often residing in a different thread block/SM.

This solution enables utilization of all the SMs and significantly increases the possible length of the simulated instrument without affecting the performance. In other words, as long as the GPU can accommodate the desired number of Cosserat

elements, the computation time is nearly constant, regardless of the length of the simulated rod. Figure 6.6 summarizes the algorithm used, highlighting the functions requiring data exchange with other Cosserat elements often residing in a different thread block/SM.

6.3.3 Collisions and self-collisions

A collision detection scheme running serially on the CPU or parallely on the GPU was implemented and described previously in section 3.1.4. An axis-aligned bounding box (AABB) hierarchy guides the broad-phase collision detection stage, and a brute-force approach is used for the narrow-phase. The AABB tree is pre-computed on the CPU and the resulting arrays containing the mesh triangles and the tree structure are copied to the GPU global memory at the initialization stage. During the broad-phase, the tree is searched iteratively using a small stack residing in the shared memory. Its size depends on the maximum depth of the tree, which is usually between 10 and 15 levels for complex anatomical models. In order to reduce thread divergence, the broad-phase reports the indices of intersecting bounding boxes and stores them in another small stack. In the narrow-phase, a brute-force collision check against the triangle(s) in the reported bounding boxes is performed. Next, a collision response vector is calculated as a weighted average of all normal vectors of colliding triangles. The weight depends on the penetration depth for the given triangle. The GPU collision detection can be used on its own, launched as a separate kernel, or invoked from within the simulation loop running on the GPU.

In the case of self-collisions or collisions with another rod, a GPU implementation of a solution presented in section 3.1.4 was used based on bounding spheres wrapping a number of neighbouring mass-points.

6.4 RESULTS AND DISCUSSION

The performance (computational times in ms) achieved on a desktop and a laptop computer with mobile GPU chipset is now discussed. The desktop was a HP x4600 workstation (Intel Core2 Quad @2.66 GHz, 8GB RAM) with NVidia GeForce GTX 560 (7 SMs @1.66GHz, 336 CUDA cores total, 1GB DRAM) running 64bit Linux (Ubuntu 12.10). The laptop was an ASUS N55SL laptop (referred onwards as ASUS, Intel Core i7 @2.2 GHz, 6GB RAM) with NVidia GeForce GT 635M (3SMs @1.35GHz, 144 CUDA cores total, 2GB DRAM) running Win7 x64. The testing environment was developed in Java7 using the *JCuda* wrapper to communicate with CUDA C and the *JME3* graphics engine for visualization. All the tests use the inextensible CoRdE modification and constraints framework explained in Chapter 3 and subsequent chapters. Please refer to Sections 3.2.2 and 5.4.2 for rod stretch and compression figures while applying distance constraints locally.

It is worth noting that the HP desktop was equipped with a single GPU that had to switch the context between OpenGL rendering and CUDA computing, which slightly slowed down the computations. This was not the case of the ASUS laptop, which was equipped with two separate GPUs (*NVidia Optimus*™ technology), therefore the NVidia GeForce GT635m was entirely dedicated to CUDA calculations. On the other hand, the Windows Driver Model (WDM) on Microsoft platforms adds around 0.1ms delay to each kernel launch and memory copy operations. This is why it was decided to run Linux on the HP testing set-up.

A total of five different tests were conducted. The first test (free space) demonstrates pure performance aspects of the Cosserat rod elastic forces and constraints framework calculations. The results were compared to other CoRdE implementations including the original (Spillmann and Teschner, 2007) and inextensible (Spillmann and Harders, 2010). Secondly, the performance of our parallel AABB collision detection algorithm is briefly presented. The third test analyses in detail a combination of the Cosserat rod and collision detection in a real-life example – a gradual rod insertion into a 3D anatomical vascular model.

Additionally, the results between desktop and mobile platforms are also compared. The fourth test extends the previous one by adding another rod of the same length and interactions between them in a catheter/guidewire manner, i.e. one rod is inserted into another and can move forwards and backwards within it. In the fifth and final test, the performance of the self-collisions is tested by tying two rods together using a double Fisherman's knot. The speed-up in performance achieved in all the tests is summarized and discussed.

The numbers in square brackets [] in the following charts indicate the number of physics time-steps (iterations) simulated per single physics update/kernel invocation. On the CPU this may not have much meaning as the computation time scales linearly, but for the GPU this yields significant performance gains as described in the following paragraphs.

6.4.1 Free space test

This test focused solely on the Cosserat rod elastic forces, the distance constraints and integration performance. The rod was locked at one end and released like a pendulum, swaying for a few seconds in free space. The performance of 1, 10 and 20 physics time-steps simulated per single kernel launch/physics update was measured. Each time-step internally ran 10 iterations over the distance constraints. No collision detection or self-collisions were involved. The maximum number of mass-points able to fit in our GPU (7 SMs) was 6x512=3072. One SM was intentionally left free for graphics rendering. For visualization purposes, calculation times were extrapolated up to 4092 mass-points. All the times given below for the GPU implementation are total execution times "visible" from the host (CPU) side. They include kernel launch overhead, as well as transferring position data of the mass-points back to the CPUs main memory for further processing and/or rendering.

6.4.1.1 GPU implementation performance characteristics

Figure 6.7 presents the behaviour of the GPU implementation for 1, 10 and 20 time-steps simulated per single kernel launch. For a single iteration, a 16 fold increase in

number of Cosserat elements (0.1742ms for 128 vs. 0.1942ms for 2048) results in just over an 11% increase in computation times. For higher number of iterations, this figure grows to 25% for 10 and 27% for 20 iterations. This small, yet noticeable increase is caused mainly by the inter-block synchronization mechanisms and, to a smaller extent, by global memory reads/writes to communicate between the SMs. Figure 6.7 also shows that increasing the number of physics time-steps from 1 to 10 and to 20 per kernel launch, yields an average increase in computation times of only x2.72 and x4.50 respectively. In other words, the overall average cost of a single time-step when executed on the GPU in a batch of 10 per kernel launch (0.49ms/10 = 0.049ms) was 73.2% lower than executing a single time-step per kernel launch (0.18ms/1). In a batch of 20, the overall average cost is further reduced to 77.5% (0.81ms/20). This is caused by the fact that, after overcoming the initial overhead of passing the control to the GPU and latency issues related with accessing the data in the GPU global memory (DRAM), the cost of subsequent physics iterations is significantly lower. The initial overhead is promptly compensated by the high arithmetic power of the GPU.

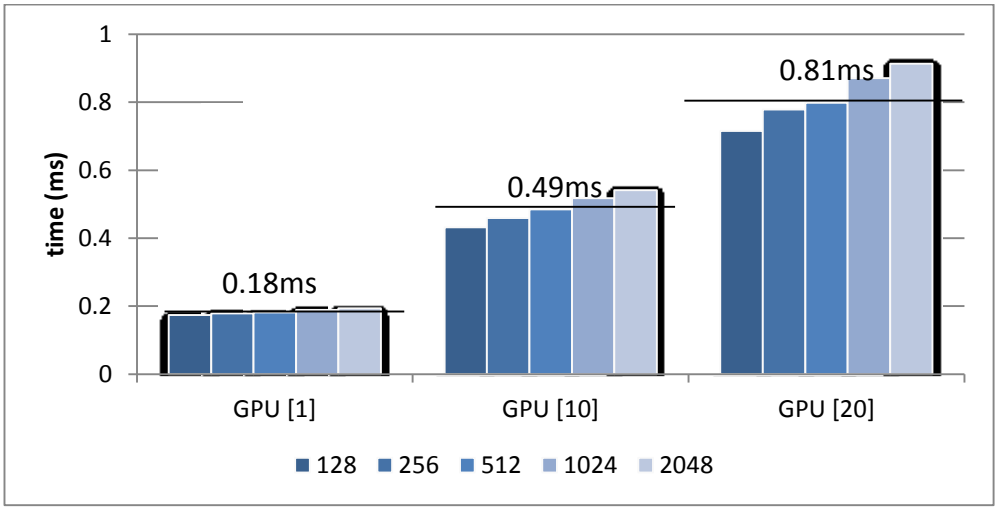


Figure 6.7: Comparison of number of iterations simulated on the GPU on the HP desktop.

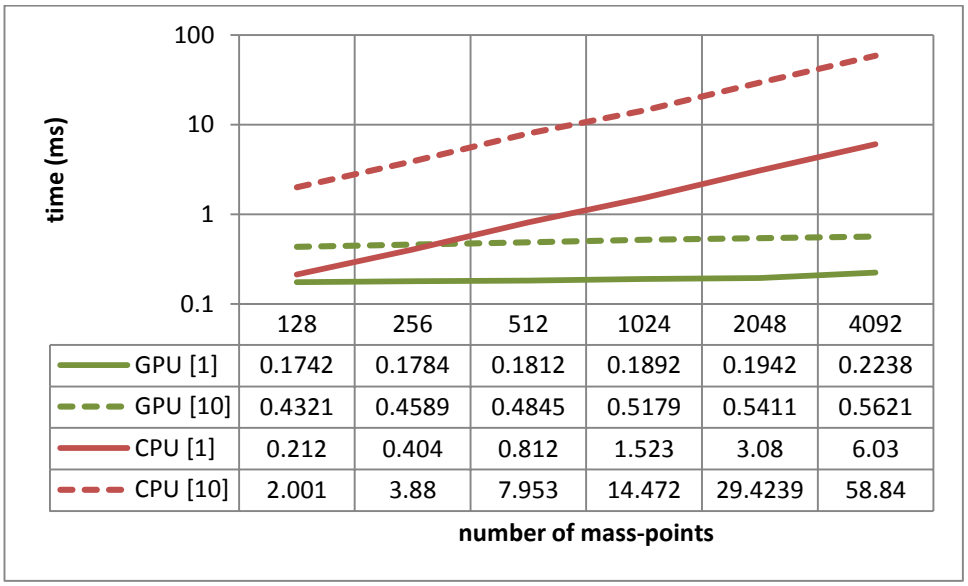
The overall average cost of a single time-step simulation when time-steps are executed on the GPU in a batch is a valid performance metric. This is because CUDA GPUs are equipped with separate and independent data copy engines. This allows

for fast, asynchronous transfers (using pinned memory and different CUDA streams) of the intermediate data between the global memories of the GPU and the CPU without waiting for the kernel to finish processing the given number of iterations. This, for example, enables writing the current state of the haptic device to the running kernel and getting back the force-feedback value at the rate higher than the rate of actual kernel executions.

The above properties enable increasing the number of simulated Cosserat elements and, at the same time, improving the accuracy of the simulation, by decreasing the time-step, at very low additional computational cost. For the CPU version, both of these metrics behaved linearly as shown in the next section.

6.4.1.2 Comparing GPU vs. CPU performance

The CUDA version was faster than an identical CPU implementation for all the number of mass-points simulated (Figure 6.8). When running a single time-step per physics update, it was x1.2 times faster even for the smallest number of mass-points (128). It achieved a speed-up of one order of magnitude for approximately 1200 mass-points, peaking at x26.9 speed-up for 4096 mass-points.



	128	256	512	1024	2048	4092
GPU [1]	0.1742	0.1784	0.1812	0.1892	0.1942	0.2238
GPU [10]	0.4321	0.4589	0.4845	0.5179	0.5411	0.5621
CPU [1]	0.212	0.404	0.812	1.523	3.08	6.03
CPU [10]	2.001	3.88	7.953	14.472	29.4239	58.84

Figure 6.8: Performance of the Cosserat rod suspended in free space without collisions in respect to the number of elements. Note the logarithmic scale on the vertical axis.

Increasing the number of iterations simulated per physics update from 1 to 10 significantly improved the results. The performance boost reached an order of magnitude for approximately 300 elements and two orders of magnitude (x100) for around 4000 elements.

6.4.1.3 Single-block vs. multi-block kernel

The above results were achieved using the most universal kernel implementation supporting inter-block synchronization. However, if the given number of Cosserat elements is small enough to fit into a single thread block, as described earlier, it is possible to discard the inter-block synchronization and communication overhead between the subsequent iterations, thus reducing calculation times even further by 35-40% (Figure 6.9). This results in speed-ups over the CPU of x10.4 (0.26ms), x19.5 (0.29ms) and x34.9 (0.32ms) for 128, 256 and 512 mass-points, respectively, compared to x6.2, x12.3, x23.0 speed-ups achieved by the universal kernel. However, on the current generation of the GPUs this is possible for maximum 512 Cosserat elements without collision detection and for 256 elements with collision detection.

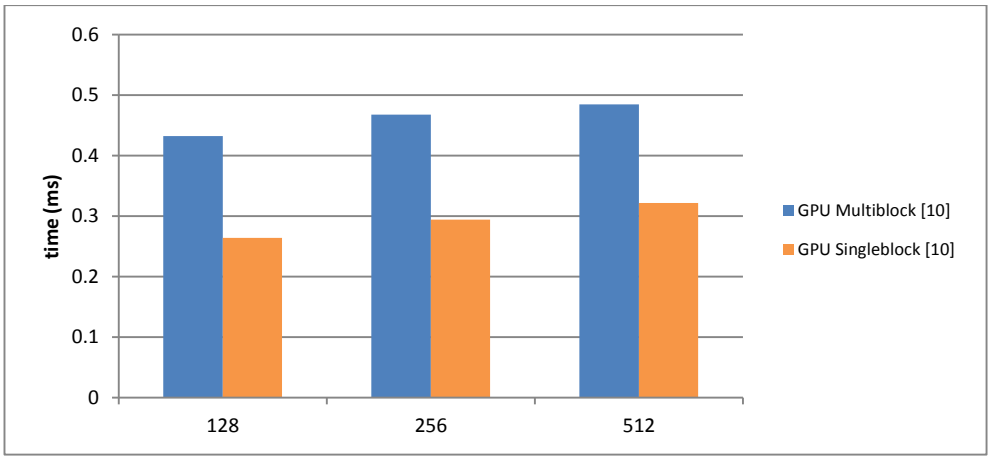


Figure 6.9: Performance of a universal, multi-block kernel vs. single-block kernel running 10 time-steps per kernel invocations

6.4.1.4 Comparing the implementation with others

Table 6.1 compares the performance of the proposed extensible and inextensible Cosserat rod implementations to others found in the literature. The inextensible,

multi-block GPU accelerated version presented in Chapter 3 was chosen as the reference. In the case of running 10 iterations per kernel launch (10 ts/kl), the overall average cost of a single time-step was used as the performance metric, i.e. dividing the total computational time by the number of iterations taken. The time measures were taken for 100 and 1000 elements running elastic force calculations, 10 iterations over distance constraints (or elastic forces calculation in case of extensible rods) and time integration, which was analogous to the previous free-space test.

Table 6.1. Performance of the extensible and inextensible Cosserat Rod GPU and CPU implementations vs. others for 100 and 1000 Cosserat elements in milliseconds.

Implementation	PC CPU	Inextensible	100 El. (ms)	1000 El. (ms)
GPU, 10 TS/KL[1]	Core2 (HP) 2.66 GHz	yes	0.43/10	0.52/10
GPU, 1 TS/KL1[1]	Core2 (HP) 2.66 GHz	yes	0.174	0.189
GPU, 10 TS/KL[2]	Core2 (HP) 2.66 GHz	no	0.26/10	0.31/10
GPU, 1 TS/KL[2]	Core2 (HP) 2.66 GHz	no	0.173	0.207
CPU, 1 TS/KL	Core2 (HP) 2.66 GHz	yes	0.198	1.783
CPU, 1 TS/KL[2]	Core2 (HP) 2.66 GHz	no	0.160	1.436
Original CoRdE (Spillmann and Teschner, 2007)	Xeon 3.8 GHz	no	0.131	1.240
Inextensible CoRdE (Spillmann and Harders, 2010)	Core2 3.0 GHz	yes	5.65*	56.5*
Simplified CoRdE (Punak and Kurenov, 2011a)	Core i7 2.93 GHz	no	0.059	0.545
Discrete Elastic Rods (Bergou et al., 2008)	Core2 2.66 GHz	yes	0.45 *	4.5*
Position Based Elastic Rods (Umetani et al., 2014)	N/A	yes	3.53*	35.3*

[1] reference inextensible GPU implementation as described in Chapter 3

[2] equivalent implementation of the original extensible CoRdE (Spillmann and Teschner, 2007)

** approximated and interpolated times as explained in the text*

Both the extensible and inextensible serial CPU implementations were slower than the original CoRdE (0.160ms and 0.198ms vs 0.131ms) by (Spillmann and Teschner, 2007), and significantly slower than the simplified CoRdE (0.160ms and 0.198ms vs 0.059ms) by (Punak and Kurenov, 2011a).

The extensible GPU CoRdE implementation was x1.68 faster than the inextensible (reference one) and x40.0 faster than the original CoRde (Spillmann and Teschner, 2007) (10ts/kl). Running a single iteration per kernel launch still yielded a decent speed-up of x5.99 of the corresponding extensible CoRdE implementation over the original one.

For 1000 Cosserat elements, our inextensible GPU implementation outperformed the original extensible CoRdE (Spillmann and Teschner, 2007) by a factor of x23.85 and the simplified CoRdE (Punak and Kurenov, 2011a) by an order of magnitude (x10.5) running 10 time-steps per kernel launch, and by x6.56 and by x2.88, respectively, running 1 ts/kl. Comparing our GPU implementation to the inextensible rods such as (Spillmann and Harders, 2010), (Bergou et al., 2008), (Umetani et al., 2014) yields speed-ups ranging from x23.8 to x186.6 running 1ts/kl and from x86.9 to even x681.5 when running 10ts/kl.

Note that different test platforms were used and that the times for 100 and 1000 mass-points in these papers are not given explicitly. In the case of (Spillmann and Harders, 2010), we derived the time from the coil embolization example where authors state constituent times of a simulation of 40 Cosserat mass-points. Adding these and linearly extrapolating from 2.26ms per 40 points gives an approximated time of 5.65ms for 100 points. In the case of (Bergou et al., 2008), we linearly extrapolated times for test 7 (0.34ms / 75 points). Umetani et al. (Umetani et al., 2014) gives the computation times just for one test (1.06ms / 30 points). Still, the models proposed in these publications have other advantages. For instance, longer computation times are compensated by higher stability and larger time-steps.

6.4.2 Collision detection

This test focused on the performance of the collision detection exclusively. No physics was simulated on the GPU. The rod was inserted into a 3D reconstruction of a vascular silicon phantom model consisting of 26066 triangles represented by an AABB tree of maximum depth of 15, consisting of 46205 nodes and 23103 leaves (Figure 6.10).

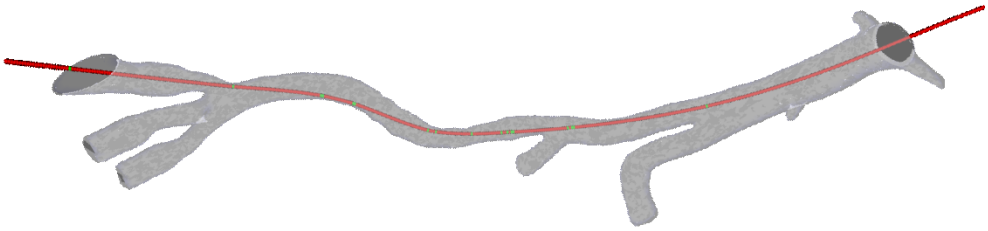


Figure 6.10: Cosserat rod consisting of 512 elements inserted into a 3D reconstruction of a vascular silicon phantom model.

On average, there were 1.12 triangles per leaf. The collision detection kernel described earlier was used. Before launching the kernel, an array containing the positions of all the mass-points had to be transferred to the GPU memory and, after the kernel execution, an array of collision response vectors was copied back to the CPU memory, which added a substantial overhead as shown in Figure 6.11.

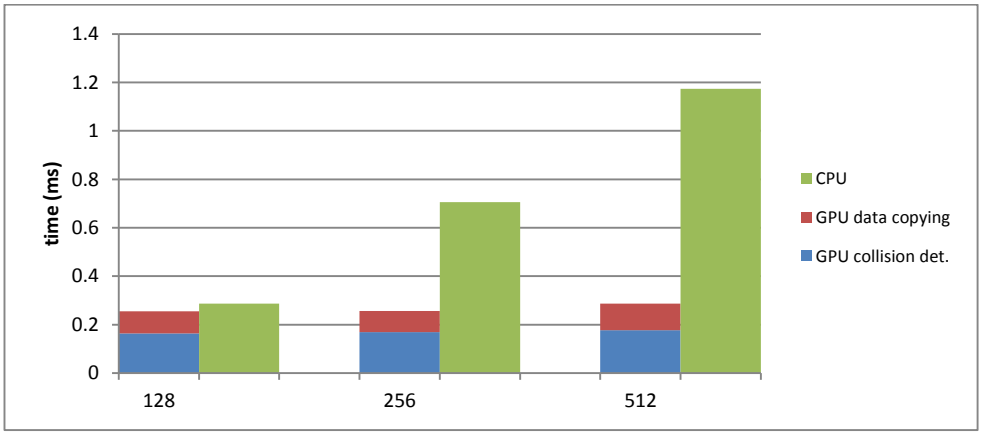


Figure 6.11: Collision detection performance comparison

Figure 6.11 illustrates that there was not much gain from using the GPU over the CPU for 128 mass points. However, as the CPU computing times grew linearly with

the number of mass-points, the GPU suffered almost no penalty. For 512 points, it was x4 times faster than the identical algorithm running on the CPU (0.29ms vs. 1.17ms). The 3D model used in the simulation was not long enough to fully accommodate rods longer than 512 elements. In a longer model of a complete aorta (18k triangles, 36k nodes, 18k leaves) with 3072 mass-points colliding (6 thread blocks, 512 mass-points each), performance on the GPU was more than x20 times faster than on the CPU.

6.4.3 Guidewire insertion test

A single instrument consisting of 512 Cosserat elements was gradually inserted into the reconstructed 3D phantom vascular model, same as described above and shown in Figure 6.10. Such number of elements is sufficient to simulate with high accuracy a guidewire navigating from the femoral artery to the heart vessels (Chapter 5), or a flexible endoscope inside a colon (Chapter 4). The rod did not self-collide. Each physics time-step contained a broad and narrow phase of collision detection, identical to the one described in the previous section, and 10 iterations over distance and collision constraints including Coulombian friction. Figure 6.12 shows that the computation times largely depended on the instrument insertion depth due to the increased number of AABB tree queries. This held true for both CPU and GPU and resulted in a performance drop of up to a half when compared to the zero insertion depth.

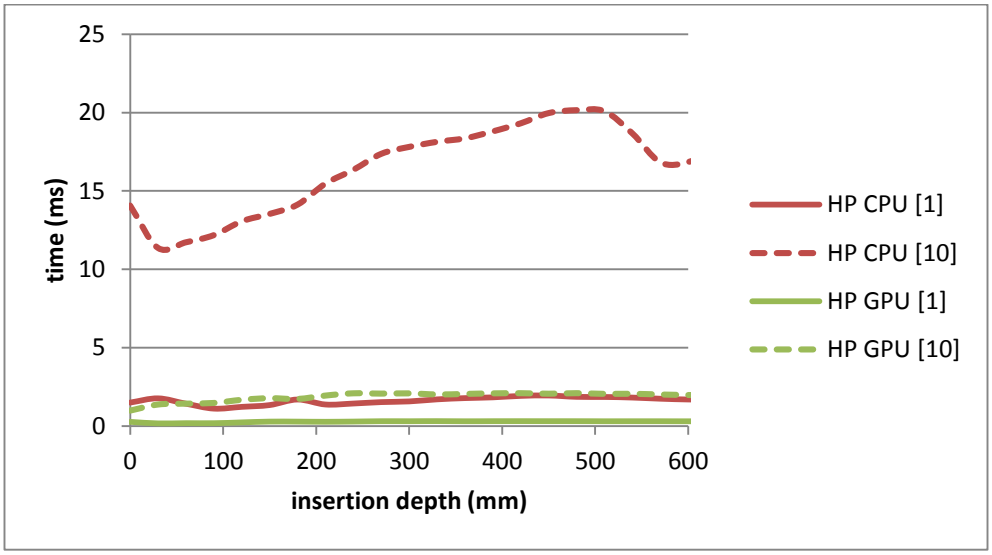


Figure 6.12: Calculation time at different insertion depths for 512 mass-points, 1 and 10 iterations per physics update/kernel launch

The average computation time throughout the insertion procedure for CUDA implementation on the HP desktop running 10 iterations per kernel launch was 1.88ms (min. 0.99ms, max. 2.11ms) and was, on average, x8.75 times faster than the CPU version (avg. 16.45ms, min. 11.37ms, max. 20.15ms). A single time-step per kernel launch running on the GPU (0.29ms) was still x5.65 times faster than the CPU (1.64ms). The overall cost of a single GPU time-step, while executed in a batch of 10 per kernel launch (1.88ms/10), was 35% cheaper than launching a single physics iteration (1.64ms/1).

Figure 6.13 shows the performance of the ASUS laptop compared to HP desktop workstation. The GPU implementation on ASUS achieved an average x5.75 speed-up since the laptop has a slower and smaller GPU (lower GPU clock rate, 3 SMs vs. 7 SMs on the HP), but a newer and faster CPU. Nevertheless, the GPU implementation on the laptop (2.49ms) was running only 25% slower than on the desktop (1.88ms, 10 ts/kl).

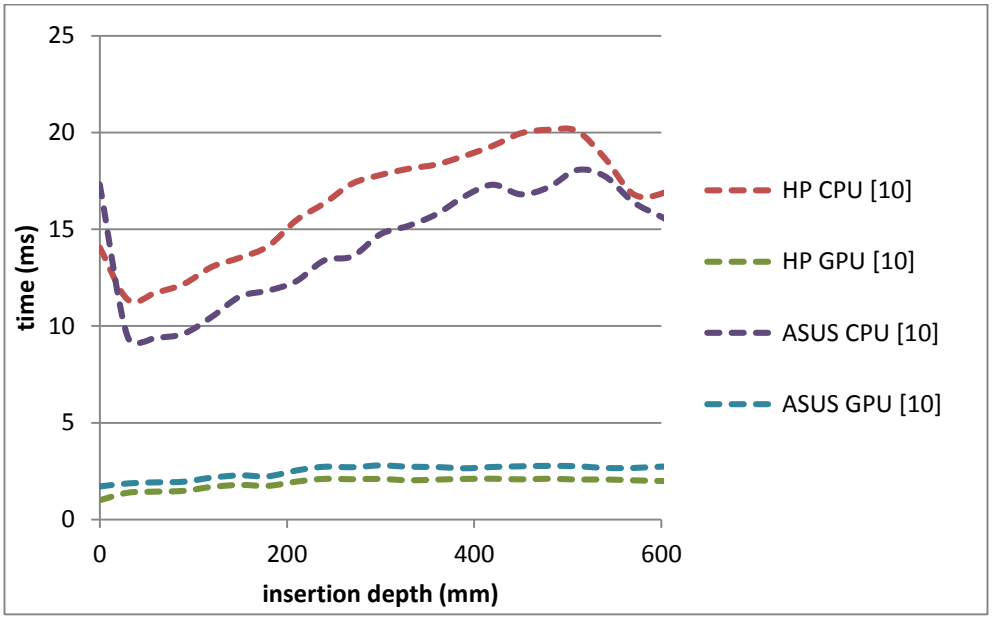


Figure 6.13: Calculation time at different insertion depths for 512 mass-points on desktop HP and laptop ASUS, 10 iterations per physics update/kernel launch

6.4.4 Catheter and guidewire insertion

This test is similar to the previous one, but with two rods inserted interacting with each other. Each rod consisted of 512 Cosserat elements. As before, each kernel invocation processed 1 or 10 time-steps per physics update, each step containing broad and narrow phases of collision detection and 10 iterations over distance and collision constraints. Figure 6.14 summarizes the results.

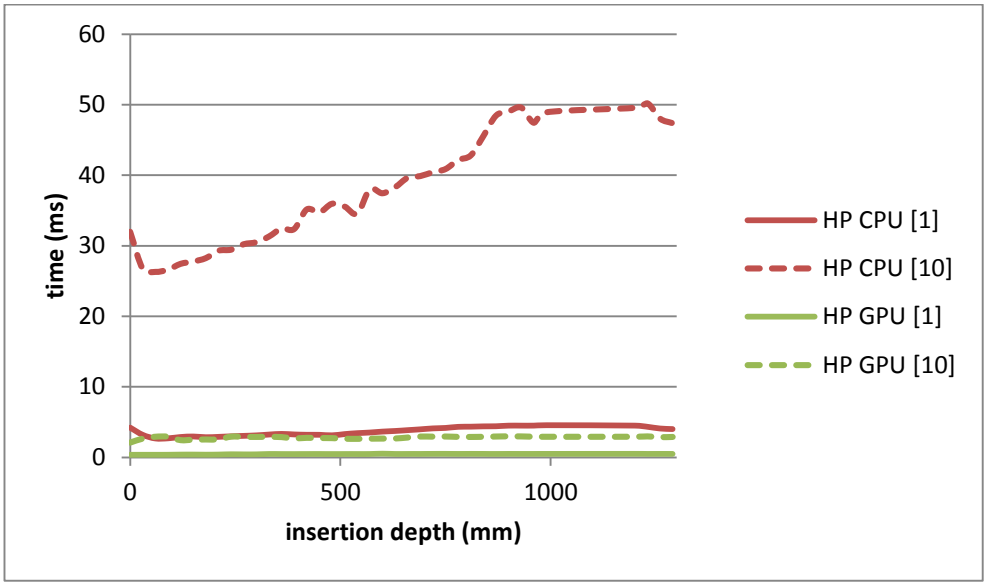


Figure 6.14: Insertion performance of two rods with bilateral interactions (v2) running 1 and 10 time-steps per kernel launch

Two variants of instruments interaction were implemented. In variant one (v1), the instruments interaction was one-way only, i.e. one of the instruments was fully dependent on the other. This can be illustrated, for example, by a soft guidewire inserted into a stiff catheter. The guidewire influence on the overall catheter-guidewire pair dynamics is negligible. In a slower, but more realistic variant two (v2), the interaction between the two rods was bidirectional with each instrument influencing the dynamics of the catheter/guidewire pair according to its mechanical properties. It is worth noting that the difference between these two variants was quite significant for the GPU version, but practically irrelevant in the CPU implementation. This is because there is more intense synchronization and communications between the blocks in a constraints loop. In the case of the binding constraints (see section 5.3.2), especially for v2, the algorithm needs to exchange data about all the Cosserat elements of the other rods via slow global memory. For comparison purposes, the performance of a variant zero (v0) that involved no interactions between the instruments was also considered. The rods were independent and did not collide with each other.

The average computation times for the CUDA implementation on the HP desktop were 1.96ms, 2.15ms and 2.80ms for variants 0, 1 and 2, respectively. The interactions (binding constraints) added 10% (v1) and 43% (v2) overhead for the given number of elements (2x512) compared to practically no overhead on the CPU. Nevertheless, the GPU implementation was x19.3 (v0), x17.6 (v1) and x13.5 (v2) times faster than variant 2 running on the CPU (37.88ms, Figure 6.15).

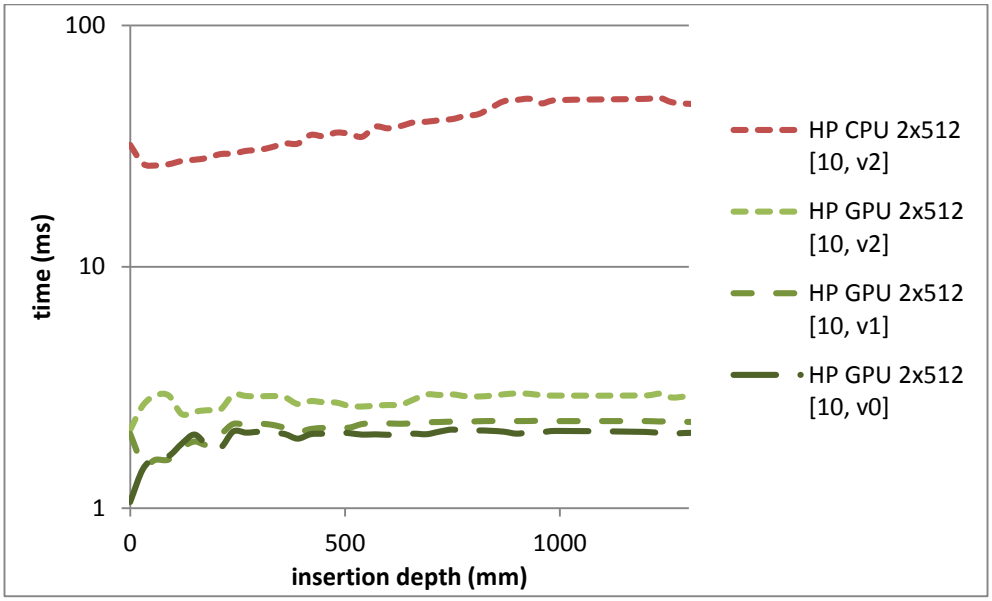


Figure 6.15. Insertion performance of two instruments using different variants running 10 time-steps per kernel launch. Note the logarithmic scale on the vertical axis.

Figure 6.16 presents variant 1 and 2 running a single time-step per kernel launch. In this case, the performance difference between these two running on the GPU was negligible (v1 - 0.44ms, v2 - 0.46ms). On average, they were x8.0 times faster on the GPU than on the CPU (3.68ms). The overall cost of a single time-step while executing a batch of 10 iterations per kernel launch (2.8ms/10) was nearly 40% less than launching a single physics iteration (v2).

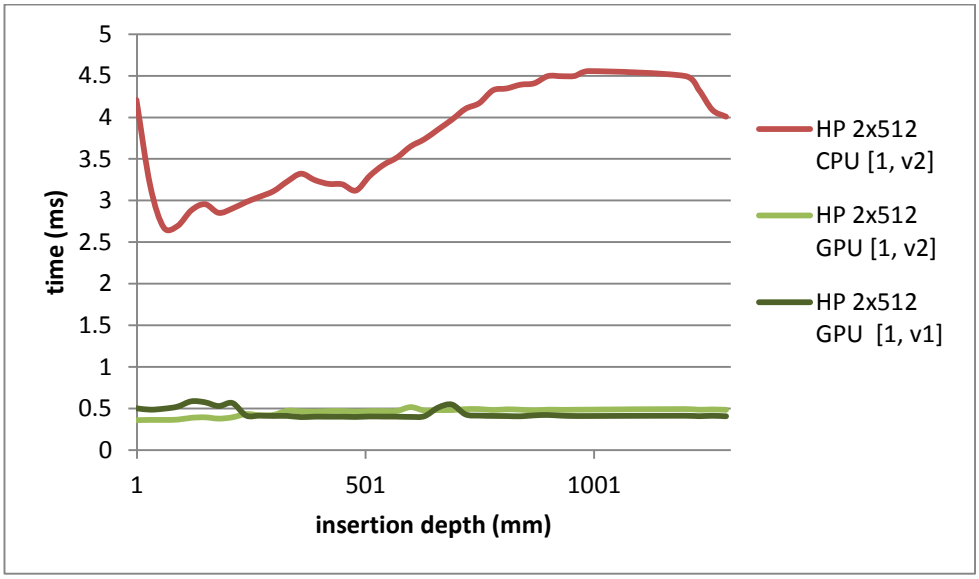


Figure 6.16: Insertion performance of two instruments using different variants running 1 time-step per kernel launch.

6.4.5 Double Fisherman's knot tying test

Figure 6.17 shows two stages of tying a double Fisherman's knot similarly to the test found in section 3.2.3 and in (Spillmann and Teschner, 2008). Two Cosserat rods consisting of 256 elements each were attached at different points in space at the ends. Two Phantom Omni haptic devices (www.geomagic.com) were used to control the loose ends and manually tie the knot. The average number of collisions for the tied knot was 300 colliding sphere-sphere pairs. A broad-phase based on bounding spheres as described in section 3.1.4 on collision detection was employed. The total calculation time on the GPU was 0.32ms for a single iteration per kernel launch and 1.72ms for 10 iterations. On the CPU, these numbers were 2.67ms and 21.99ms, giving speed-ups of x8.34 and x12.78, respectively. Hence, enabling for use at haptic interactive rates.

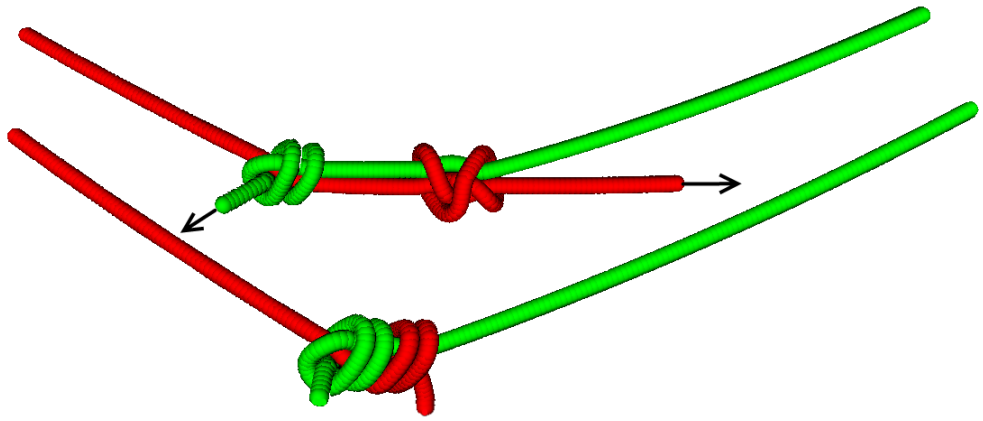


Figure 6.17: Tying a double Fisherman's knot

The double Fisherman's knot was chosen since it has been previously used in a paper on knots simulation using the CoRdE model (Spillmann and Teschner, 2008). Although the overall approach to knot simulation presented here is different, the implementation is x41.5 faster on the GPU (0.32ms compared to 13.3.ms) and x5.0 faster on the CPU (2.67ms compared to 13.3.ms). Even comparing to the fastest adaptive rod version reported in (Spillmann and Teschner, 2008) (7.3ms), the GPU implementation still attains a 20x speed-up. This was in spite of having to use twice as many Cosserat elements to be able to tie this knot and keep it stable, due to the self-collision detection being based on spheres rather than actual centreline geometry, which required increasing the number of spheres to make them densely overlap each other (Figure 6.17).

Comparing the results to the suture simulation based on a fast, simplified CoRdE by Punak et al. (Punak and Kurenov, 2011a), the inextensible GPU solution was x7.84 faster than their serial version (0.32ms compared to 2.51ms) and x17.9 faster than their CUDA attempt (0.32ms compared to 5.73ms) for the same number of Cosserat elements (512).

6.5 SUMMARY

This chapter has presented a CUDA-based massively-parallel implementation of a Cosserat rod simulation framework and its modification to ensure inextensibility. By superseding the *CUDA Scalable Programming Model* and using inter-block synchronization, it was possible to simulate multiple physics time-steps per single kernel launch utilizing all the GPU streaming multiprocessors. Under some constraints, this results in nearly constant computation time, regardless of the number of Cosserat elements simulated. Moreover, improving the simulation accuracy by decreasing the time-step size and increasing their number, results in relatively low additional computational cost.

Comparing the results against other Cosserat rod implementations speed-ups of at least an order of magnitude were obtained when simulating 1000 Cosserat elements on a consumer level GPU, when running 10 iterations per single CUDA kernel launch. The extensible GPU CoRdE implementation was x40.0 faster than the original CoRdE version.

In a series of tests, the inextensible Cosserat rod achieved an average speed-up of x15.11 running 10 time-steps per kernel launch, and an average speed-up of x7.32 running a single time-step per kernel launch over the corresponding CPU version for a moderate number of Cosserat elements (512-1024, Figure 6.18).

The first test, a free-space test, showed an interesting performance characteristic of the GPU implementation, namely that of nearly constant computations times. Adding collision detection allowed for more real-life test scenarios such as simulating a guidewire/catheter pair navigated in a vascular system. In the last test, self-collisions were added and it was shown that the proposed solution can also be used to efficiently simulate sutures by manually tying a double Fisherman's knot. The collision detection clearly was a performance bottleneck for both CPU and GPU.

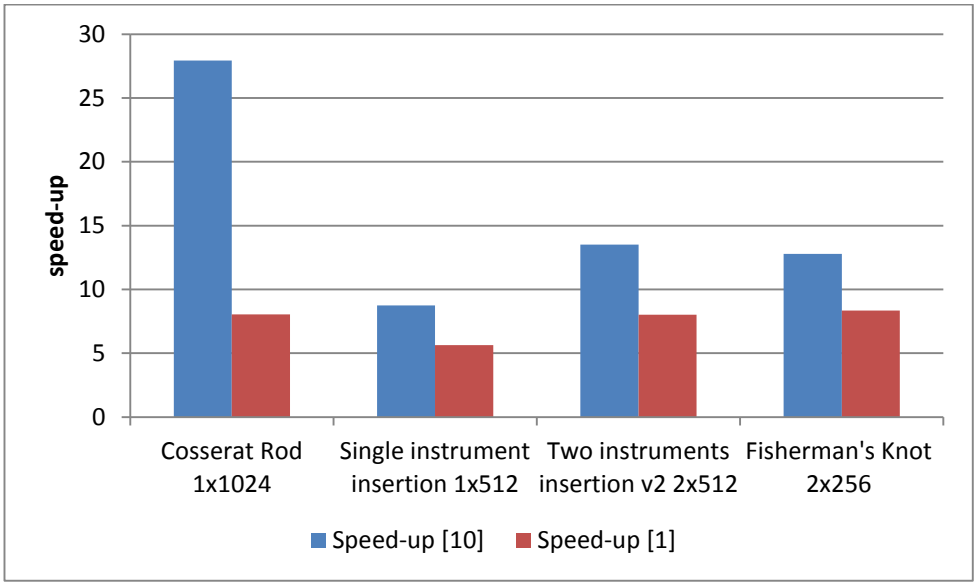


Figure 6.18: Speed-ups of our GPU implementation achieved over corresponding CPU version in different test sets running 1 and 10 time-steps per single kernel launch.

The speed-ups achieved enabled for accurate real-time simulation at haptic interactive rates (0.5-1kHz), even when executing a single physics time-step per kernel launch. However, due to time constraints, the GPU implementation was not fully integrated and validated with the VCSim3 system presented in the previous chapter. The expectation is such integration will improve the behaviour of virtual instruments and enable for the simulation of catheter/guidewire pair reaching all the way from the femoral artery to the coronaries

In summary, the presented GPU accelerated model is an interesting choice for fast and accurate elastic rod simulation, not only for medical applications, whilst demonstrating that, for some applications, the GPU can deliver a significant performance boost, even when the problem size is relatively small (128-4096 Cosserat elements) in terms of traditional massively parallel computation standards.

Chapter 7

CONCLUSIONS

This chapter summarizes the work and research carried out in the field of virtual reality simulators for minimally invasive surgery training using flexible instruments. The main objective was to choose and implement an elastic rod model and validate its realism so that it could be adapted to accurate real-time simulation of virtual flexible surgical instruments. This aim was achieved through the development of two prototype VR simulators using a proposed inextensible Cosserat rod model for the simulation of flexible tools. NOViSE is the first flexible endoscopy VR simulator for Natural Orifice Transluminal Endoscopic Surgery (NOTES). VCSim3 is a VR simulator for endovascular interventions relying on catheters and guidewires. The realism of the simulation was methodically validated and received favourable feedback from clinicians, both in terms of the behaviour of the flexible instruments, as well as the simulator prototypes as a whole. Furthermore, following the recent trends in the field, an efficient massively-parallel CUDA implementation of the Cosserat rod was proposed.

7.1 SUMMARY OF WORK AND ACHIEVEMENTS

7.1.1 To review software and hardware components necessary to develop a modern computer based virtual reality medical simulators

In Chapter 2, an overview of the components of a modern computer based medical simulator was presented placing particular emphasis on the physically based simulation aspects. This allowed for a better understanding of both hardware and software requirements of such systems and to select the suitable algorithms and development tools for the project. The main decisions made during this planning phase were to:

- implement the simulators in Java language
- use an existing open-source solution for 3D graphics rendering
- implement own physics algorithms rather than using third-party physics engines or simulation frameworks

Considering the application field, the use of Java as the implementation language can be, admittedly, for some quite surprising. The initial tests with basic algorithms indicated, however, that the performance of Java can nearly match the speed of the lower level unmanaged languages such as C/C++. This is because, after an initial "warm-up" consisting of processing a few simulation loop iterations, the computationally intensive "hot spots" were successfully identified and compiled "just-in-time" into the native code by the Java's virtual machine. If, for some reason, that was not the case, the particular function was implemented in C/C++ and invoked via the Java Native Interface (JNI). Automatic memory management was very helpful to speed-up the development by eliminating memory leaks and other hard-to-detect memory-related bugs. The possibility to edit and "hot-swap" the code on the fly during the runtime with a 3D live preview was extremely handy, especially when debugging unstable and potentially exploding physics settings. Moreover, in the author's personal opinion, the integrated development tools (IDEs) and profilers for Java such as Eclipse (www.eclipse.org) are, in some aspects,

superior to the ones of other languages. In summary, Java turned out to be an apt choice for prototyping. However, in order to ensure the highest computational performance, the production versions should be fully implemented in C/C++.

Using an existing 3D graphics framework was clear from the beginning. At the time of the project start, there was a range of decent open-source engines and implementing own solution would be clearly "rediscovering the wheel". Use of Java narrowed the possibilities to a couple of options from which the JME3 seemed the most mature, actively developed and supported by the community. The possibility to participate in the improvement of an open-source project such as JME3 by reporting and fixing bugs was another advantage over the closed-source solutions.

The JME3 has a built-in support for the well-known Bullet Physics library, either via its direct Java port (www.jbullet.advel.cz) or a C/C++ wrapper to the native library. However, after conducting some initial experiments, it was decided not to use it. This was because Bullet is mainly a rigid bodies engine and, at the time, it offered only experimental support for deformable bodies. Other well-established physics solutions such as PhysX or Havok had similar downsides, plus they were not open-sourced.

The use of dedicated frameworks for medical simulators such as SOFA was carefully considered. However, the initial learning curve was deemed too steep. Thus, it was decided to have full-control over the code, develop a custom framework from the ground-up, and thus learn considerably more in the process. Admittedly, implementing mesh importers, scene serialization methods or basic data structures for collision detection added a substantial overhead, but the gained know-how and expertise turned out to be very valuable in the long run from both an algorithmic and software engineering perspective.

7.1.2 To review models of one-dimensional flexible bodies and choose the most appropriate in terms of realism and performance

In the second part of Chapter 2, existing models of one-dimensional deformable bodies were reviewed. The choice for interactive solutions spans a number of

application fields, most notably, hair simulation and a couple of models adapted for catheter and guidewire simulation. Choosing the right model was difficult as the publications derive from many distinct fields such as mechanical engineering or computational geometry, to name just a few. The authors usually do not share or disclose their reference source code. Thus, an efficient, artefact and bug free implementation of such models is a non-trivial and time consuming task.

At first, experiments with a mass-spring model that had previously been developed in the group (Luboz et al., 2009b) were carried out, but the drawbacks of such methods quickly became apparent, especially when torsion was considered. The solutions based on Cosserat theory were the next choice. The models based on reduced coordinates formulation were rejected due to known difficulties in collisions and self-collisions handling. Instead, the model with an explicit centreline representation was chosen which, according to its authors, enables for simulation of complex contacts and looping phenomenon. This was a desired characteristic for one of the clinical applications under consideration (NOTES). The CoRdE model (Spillmann and Teschner, 2007) seemed to be fast, dynamic and elegant, as well as reasonable in terms of implementation complexity. Moreover, the rod physics calculations were processed locally, which made it a good candidate for a prospective massively-parallel implementation.

7.1.3 To adapt the chosen model to fit the requirement of the software framework under development

The simulation of the virtual elastic rod is determined by a set of customizable parameters affecting its speed and accuracy. Finding the right balance between these two factors is a challenging problem requiring some compromises. As stated in (Spillmann and Teschner, 2007), the mechanical parameters of the virtual rod do not directly correspond to real world values. Thus, the next step after implementing the model, was to derive parameters which, first of all, met the requirements for a stable real-time simulation at haptic interactive rates (1 kHz) and, at that initial stage, allowed for mechanical parameters recreating the behaviour of instruments

ranging from thin and light guidewires to much heavier and thicker flexible endoscopes.

One of the major problems encountered with the CoRdE model was the compressibility of the rod linearly proportional to its length and quadratically to the time-step. From the user perspective, it appeared as a noticeable delay in the movement of the rod distal end during the manipulation. This was due to the stretch energy used in the CoRdE model, which was in fact acting as a penalty method linking neighbouring mass-points. The semi-implicit integrator required an unacceptable small time-step for the rod to stay responsive. The penalty method was therefore replaced by a fast block iterative constraint solver guaranteeing rod inextensibility.

For the collision detection, a custom scheme based on AABB hierarchy was implemented, which could efficiently handle dynamic deformable objects. As the constraint solver advanced, the collision response was also altered. The penalty method was abandoned and replaced with an iterative impulse-based approach added to the constraint solver.

Summarizing, the presented Cosserat rod implementation enabled for efficient modelling of bending, twisting and (non)-stretching phenomena, easy parametrization, as well as guaranteeing almost immediate response to user manipulations, even for long instruments. The implementation runs efficiently on an off-the-shelf PC or laptop, significantly exceeding the minimum required haptic interactive rate. However, achieving haptic rates for very computationally intensive scenarios such as a catheter/guidewire pair reaching all the way from the femoral artery into the heart coronaries, was problematic. Therefore, a novel massively-parallel GPU implementation was presented in Chapter 6.

7.1.4 To apply the chosen model to develop a virtual reality simulator for flexible endoscopy, specifically, for NOTES

After implementing, fine-tuning and testing of the initial elastic rod model, it was adapted for the simulation of a flexible endoscope. Next, a complete hybrid trans-gastric cholecystectomy procedure was recreated under the supervision of NOTES experts, followed by the validation studies.

An initial verification of the NOViSE simulator was carried out by obtaining subjective feedback (face and content validity) from 14 clinicians of different specialities. Four of them were qualified as NOTES experts who have independently performed 10 or more human or animal-model NOTES procedures. The construct validity was established by comparing a range of performance metrics between the expert group and other clinicians (novices).

NOViSE exhibited good overall face and content validity. Participants were especially convinced by the movement and looping phenomenon of the virtual endoscope. The visual side of the simulator also scored high grades proving that a free open-source graphics engine can nowadays deliver realistic visualization for medical simulation, even on a consumer-level mobile GPU. The static, undeforming meshes of the abdomen anatomy and the manually tuned mass-spring model chosen for the gallbladder deformation, although relatively simple compared to the state-of-the art, were sufficiently realistic for the majority of the participants.

In terms of content validity, participants largely agreed that NOViSE is a useful training tool for NOTES and that they would recommend it to others. Regarding construct validity, a trend indicating that experts were faster and used a shorter path length than novices in all but one task was observed.

Summarizing, NOViSE laid promising foundations for further development. The Cosserat rod model turned out to be an apt choice for simulating a flexible endoscope in NOTES procedures. NOViSE allowed for the identification of problems with hardware and to raise a series of questions regarding the simulation software. It also allowed to assess, together with clinicians, which improvements and

extensions would benefit the future version the most. NOViSE caught the attention of a surgical hardware innovation group who enquired about the feasibility of using the simulator to facilitate the development of an extension for real, advanced flexible endoscopy.

7.1.5 To apply the chosen model to develop a virtual reality simulator for endovascular interventions

In order to assess the Cosserat rod model usability for simulation of flexible instruments of very different mechanical properties, the VCSim3 – a VR simulator for endovascular interventions was developed. First of all, this required recreating the interactions between the virtual catheters and guidewires, as well as rapidly moving coronary vessels. Next, a range of supplementary solutions such as fluoroscopic visualization, contrast flow propagation, balloon and stent deployment was developed allowing for simulation of a complete angioplasty and stenting procedures under the supervision of senior interventional cardiologists. Then, the mechanical parameters of six guidewires and three catheters were optimised with respect to their real counterparts scanned in a silicone phantom using CT. The results exhibit near sub-millimetre accuracy.

An initial verification of the simulator was carried out by obtaining subjective feedback (face and content validity) from 17 cardiologists. The results of the face validity suggest that VCSim3 demonstrated early signs in terms of the realism of the simulated instruments. Nearly half of the participants was positive about the behaviour of virtual catheters and guidewires. The results of the face validity of supporting solutions such as fluoroscopic visualization, cardiac motion, contrast propagation, balloon inflation and stent deployment ranged from neutral to positive. As in the case of the NOTES simulator, this suggests that, although technically simple, the supporting solutions were sufficiently realistic to convince the majority of participants. This raises the question if more complex solutions are indispensable in order to deliver an effective training experience in virtual reality.

In terms of content validity, more than half of responses were positive or very positive indicating that the majority of the participants agreed that VCSim3 is a useful training tool for endovascular interventions, and that they would recommend it to others.

7.2 LIMITATIONS AND FUTURE WORK

The presented work has a number of limitations. These are identified and discussed below, together with suggestions for potential improvements.

7.2.1 Elastic rod model

Although the inextensible Cosserat Rod model worked well for the chosen applications, it is an approximation to the real-world phenomenon requiring a trade-off between accuracy and performance. The main limitation of the model is the penalty method used for the parallel constraint aligning material frames to the centreline. Such an approach is fast and simple, but adds additional stiffness to the system and an extra parameter to tune. Enforcing the parallel and distance constraints by using an analytical method was presented in (Spillmann and Harders, 2010). However, it requires solving a banded system governing both mass-points and material frames, which is less efficient and complicates the parallelization of the model.

Even though the presented constraint solver guarantees nearly perfect rod inextensibility and practically immediate response to linear manipulations, this is not the case for rod rotations. The latency during rotations can be noticeable, especially in the case of longer rods and curved rods.

Finally, it would be interesting to compare the presented implementation in terms of performance, stability and accuracy to the other models based on Cosserat theory using a common framework and a set of test cases.

7.2.2 NOViSE VR

Owing to the fact that NOViSE is a prototype and limited time available for its development, the simulation software also has restrictions. First of all, only one procedure (cholecystectomy) and one approach path (trans-gastric) is currently available. Therefore, supporting a wider range of procedures (e.g. appendectomy) and approaches (trans-vaginal, trans-rectal) is an obvious next-step in its development.

During the virtual procedure, the fat surrounding the Calot's triangle is lacking and the steps of creation and closure of the viscerotomy site are omitted. Adding these features and tasks is necessary to recreate a broader NOTES experience. However, this requires much more work in terms of the deformation of organs and their topological modifications

The soft-body model used for gallbladder simulation is much simpler than the current state-of-the-art. The other organs are static and do not deform. It would be interesting to examine if more advanced deformation models such as FEM and/or simulating a deformable throat, stomach, liver, etc. would make a perceptible difference to the user experience in the context of a full procedure.

During the pilot study, participants mainly complained about the haptic device stating that the physical shaft was too stiff and heavy, and about the poor ergonomic design of the handpiece. The linear travel length of the scope is also, at least, sub-optimal and the necessity to reposition the endoscope at certain points of the procedure might be disturbing for some users. These are fortunately problems solvable in a "Mark II" version of the haptic device. However, it is suspected that current hardware issues might result in "the break in presence" and could divert the user from noticing the deficiencies of the simulation software. It would be interesting to see if solving those problems translates to a more critical assessment of the simulation software.

The validation study, although indispensable to identify the main problems with the simulator, suffered from a small number of participants, particularly, NOTES experts who could provide more valuable feedback. As a result, a second round of validation is foreseen after implementing the identified modifications to both hardware and software components of NOViSE.

7.2.3 VCSim3

Similarly to NOVISE, VCSim3 also suffered from the restrictions of the haptic device (listed in section 5.4.6.5). Unfortunately, in this case the problem is harder to solve, as the VSP device was provided by a third-party. To the best of the author's

knowledge, there are no other haptic interfaces for catheter/guidewire tracking currently available for non-commercial or academic use. An attempt was made to use the haptic device of modern commercial simulators, but without SDKs provided by the manufacturer this was not possible.

In terms of software, the number of available procedures is currently limited to angioplasty and stenting of left and right coronary arteries extracted from a single patient-specific CT dataset. Adding more cases is certainly supported, but requires a quite tedious process of animating them manually by overlying frame by frame onto the virtual beating heart model. A semi-automated method of extracting animated polygonal meshes out of the combination of 4D CT and 3D MRI scans was evaluated at the start of the project, but eventually abandoned due to unsatisfactory initial results in the available time. Considering the rapid improvements in medical image processing, this approach may be worth re-evaluating.

The catheter and guidewire behaviour, although on average positively acknowledged by the participants, met up with more mixed feedback than the virtual endoscope in NOViSE. Therefore, more investigation is needed in order to identify some subtle phenomena occurring during the real endovascular procedures.

The supporting solutions implemented in the simulator, although visually plausible and positively rated by the participants, lag behind the state-of-the-art. For contrast medium propagation, a method based on actual fluid dynamics such as Smoothed Particles Hydrodynamics (Tan and Yang, 2009) would be more suitable. The recent advancements in this field exploiting massively-parallel computations (Goswami et al., 2010, Krog and Elster, 2012, Rustico et al., 2014) make them feasible for interactive use. Moreover, by using a unified solver (Macklin, 2014) relying on position based dynamics (Muller et al., 2007, Macklin and Muller, 2013, Umetani et al., 2014), it would be interesting to recreate bilateral interactions between the blood in vessels and the instruments. The mechanisms of balloon inflation and stent

deployment are not based on physics. The heart vessels, although moving according to the cardiac cycle, do not deform in contact with the instruments. Also, due to the discreet collision detection, the instruments occasionally slip out of the rapidly moving heart coronaries.

7.2.4 Haptic interfaces

Although this thesis largely focuses on the simulation software, the author's impression is that the hardware plays a more important role than initially expected. This becomes apparent when realizing that an average practitioner spends thousands of hours grasping and manipulating surgical instruments. Therefore, even the slightest deviation of shape or weight can severely affect the simulated experience, no matter how good and realistic the software is. Nevertheless, solving the hardware problems and designing new, robust, compact, lightweight and transportable haptic interfaces is a matter of time.

7.2.5 GPU implementation

The emergence of general purpose computation on graphical processors (GPGPU) opens new possibilities in the field by delivering speed-ups even of two orders of magnitude. In this thesis, it was demonstrated that the GPU can deliver a significant performance boost, even for relatively small problems. However, the presented GPU implementation suffers from a couple of drawbacks. First, it is limited to NVidia GPUs only. Second, the "against the rules" multi-block implementation, although very fast and thoroughly tested on four hardware platforms, may, hypothetically, not function as expected on a different GPU model, operating system, CUDA SDK or even driver version. This may manifest as an unstable or peculiar rod behaviour (inter-block synchronization problems), longer computation times (kernel launch latency issues) or even a system crash (deadlocking all the GPU blocks).

Due to performance reasons, the proposed GPU approach averts applying distance constraints in a global manner, which increases the rod compressibility and extensibility. Admittedly, this can be mitigated by an increased number of fast physics iterations, constraints solver iterations over distance constraints and a

smaller time-step. However, such a “brute-force” approach geared towards simulating as many physics time-steps as quickly as possible has its limits.

In most of the tests the collision detection accounted for a large fraction of the computational time. Due to the high GPU threads divergence, the broad-phase based on axis-aligned bounding boxes might not be the fastest one. Other collision detection schemes such as uniform grids, hierarchical grids or spatial hashing could be evaluated for the GPU architecture.

Furthermore, the overall complexity of the simulator implementation using the GPU acceleration has increased. For fast execution all the steps of the physics simulation ought to be processed inside the loop running on the GPU. Due to memory copies between the CPU and GPU this may not always be feasible or, sometimes, even possible. An alternative to the GPU implementation may be adaptive approaches (Spillmann and Teschner, 2008) with the number of Cosserat elements dynamically adjusting to the rod curvature and/or number of contacts.

Consequently, massively-parallel programming can indeed, in some applications, deliver a significant performance boost, but it is currently associated with a steep learning curve and lacks flexibility compared to a serial implementation.

7.3 OUTLOOK

The flexible surgical instruments simulated using the Cosserat rod model delivered the anticipated performance and realism. This was confirmed by the development and initial validation of two virtual reality simulators built around this model: NOViSE and VCSim3. Having met the initial aims, there are various important aspects that can be improved. However, the research and work presented in this thesis contributes towards effective computer-based training of basic manual skills for flexible endoscopy, endovascular interventions and, possibly, other surgical procedures relying on flexible instruments. Once the identified shortcomings are addressed and additional cases are included, the next iterations of NOViSE and VCSim should offer the possibility for training of intermediate skills.

REFERENCES

ACKERMAN, M. J. 1998. The visible human project. *Proceedings of the Ieee,* 86**,** 504-511.

AGGARWAL, R., WARD, J., BALASUNDARAM, I., SAINS, P., ATHANASIOU, T. & DARZI, A. 2007. Proving the effectiveness of virtual reality simulation for training in laparoscopic surgery. *Ann Surg,* 246**,** 771-9.

AHN, W., DARGAR, S., HALIC, T., LEE, J., LI, B., PAN, J., SANKARANARAYANAN, G., ROBERTS, K. & DE, S. 2014. Development of a Virtual Reality Simulator for Natural Orifice Translumenal Endoscopic Surgery (NOTES) Cholecystectomy Procedure. *Stud Health Technol Inform,* 196**,** 1-5.

ALDERLIESTEN, T., KONINGS, M. K. & NIESSEN, W. J. 2004. Simulation of minimally invasive vascular interventions for training purposes. *Comput Aided Surg,* 9**,** 3-15.

ALDERLIESTEN, T., KONINGS, M. K. & NIESSEN, W. J. 2007. Modeling friction, intrinsic curvature, and rotation of guide wires for simulation of minimally invasive vascular interventions. *IEEE Trans Biomed Eng,* 54**,** 29-38.

ALLARD, J., COTIN, S., FAURE, F., BENSOUSSAN, P. J., POYER, F., DURIEZ, C., DELINGETTE, H. & GRISONI, L. 2007. SOFA - an Open Source Framework for Medical Simulation. *Medicine Meets Virtual Reality 15,* 125**,** 13-18.

ALLARD, J., COURTECUISSE, H. & FAURE, F. 2012. Implicit FEM Solver on GPU for Interactive Deformation Simulation. *In:* HWU, W.-M. W. (ed.) *GPU Computing Gems Jade Edition.* Boston: Morgan Kaufmann.

ANJYO, K., USAMI, Y. & KURIHARA, T. 1992. A Simple Method for Extracting the Natural Beauty of Hair. *Siggraph 92 : Conference Proceedings,* 26**,** 111-120.

ANTMAN, S. 1995. *Nonlinear Problems of Elasticity*, Springer Verlag.

AVRIL, Q., GOURANTON, V. & ARNALDI, B. 2009. New trends in collision detection performance.

BANKS, J. 2010. *Discrete-event system simulation,* Upper Saddle River, Prentice Hall.

BARAFF, D. 1994. Fast contact force computation for nonpenetrating rigid bodies. *Proceedings of the 21st annual conference on Computer graphics and interactive techniques.* ACM.

BARAFF, D. & WITKIN, A. 1997. Physically Based Modeling (Online Siggraph '97 Course notes).

BENDER, J., ERLEBEN, K. & TRINKLE, J. 2014. Interactive Simulation of Rigid Body Dynamics in Computer Graphics. *Computer Graphics Forum,* 33**,** 246-270.

BERGEN, G. J. A. V. D. 2004. *Collison detection in interactive 3D environments,* Amsterdam ; Boston, Elsevier/Morgan Kaufman.

BERGEN, G. V. D. 1998. Efficient collision detection of complex deformable models using AABB trees. *J. Graph. Tools,* 2**,** 1-13.

BERGOU, M., WARDETZKY, M., ROBINSON, S., AUDOLY, B. & GRINSPUN, E. 2008. Discrete elastic rods. *Acm Transactions on Graphics,* 27.

BERKLEY, J., TURKIYYAH, G., BERG, D., GANTER, M. & WEGHORST, S. 2004. Real-time finite element modeling for surgery simulation: an application to virtual suturing. *IEEE Trans Vis Comput Graph,* 10**,** 314-25.

BERKLEY, J., WEGHORST, S., GLADSTONE, H., RAUGI, G., BERG, D. & GANTER, M. 1999. Fast finite element modeling for surgical simulation. *Medicine Meets Virtual Reality,* 62**,** 55-61.

BERTAILS, F., AUDOLY, B., CANI, M. P., QUERLEUX, B., LEROY, F. & LEVEQUE, J. L. 2006. Super-helices for predicting the dynamics of natural hair. *Acm Transactions on Graphics,* 25**,** 1180-1187.

BERTAILS, F. A. A., BASILE AND QUERLEUX, BERNARD AND LEROY, FREDERIC AND LEVEQUE, JEAN-LUC AND CANI, MARIE-PAULE 2005. Predicting Natural Hair Shapes by Solving the Statics of Flexible Rods. *Eurographics.*

BITTNER, J. G. T., MELLINGER, J. D., IMAM, T., SCHADE, R. R. & MACFADYEN, B. V., JR. 2010. Face and construct validity of a computer-based virtual reality simulator for ERCP. *Gastrointest Endosc,* 71**,** 357-64.

BOURGUIGNON, D. & CANI, M. P. 2000. Controlling anisotropy in mass-spring systems. *Computer Animation and Simulation 2000***,** 113-123.

BRIDGES, M. & DIAMOND, D. L. 1999. The financial impact of teaching surgical residents in the operating room. *Am J Surg,* 177**,** 28-32.

BRO-NIELSEN, M. 1998. Finite element modeling in surgery simulation. *Proceedings of the Ieee,* 86**,** 490-503.

BROWN, J., LATOMBE, J. C. & MONTGOMERY, K. 2004. Real-time knot-tying simulation. *Visual Computer,* 20**,** 165-179.

CASATI, R. & BERTAILS-DESCOUBES, F. 2013. Super Space Clothoids. *Acm Transactions on Graphics,* 32.

CATTO, E. 2005. Iterative Dynamics with Temporal Coherence. *GDC.*

CAVUSOGLU, M. C., GOKTEKIN, T. G., TENDICK, F. & SASTRY, S. 2004. GiPSi: An open source/open architecture software development framework for surgical simulation. *Medicine Meets Virtual Reality 12,* 98**,** 46-48.

CFN. 2014. *http://www.thecrowdfundnetwork.com/* [Online]. Available: http://www.thecrowdfundnetwork.com/mobile-gaming-could-drive-entire-game-industry-to-100b-in-revenue-by-2017/ [Accessed 20.02.2015.

CHAI, M., ZHENG, C. & ZHOU, K. 2014. A reduced model for interactive hairs. *ACM Trans. Graph.,* 33**,** 1-11.
CHIPPERFIELD, A., FLEMING, P., POHLHEIM, H. & FONSECA, C. 1994. Genetic Algorithm TOOLBOX For Use with MATLAB.
CHOE, B., CHOI, M. G. & KO, H. S. 2005. Simulating complex hair with robust collision handling. *Proceedings of the 2005 ACM SIGGRAPH/Eurographics symposium on Computer animation.* Los Angeles, California: ACM.
CHOI, K. S., CHAN, S. H. & PANG, W. M. 2012. Virtual Suturing Simulation Based on Commodity Physics Engine for Medical Learning. *Journal of Medical Systems,* 36**,** 1781-1793.
COLES, T. R., MEGLAN, D. & JOHN, N. W. 2011. The Role of Haptics in Medical Training Simulators: A Survey of the State of the Art. *Ieee Transactions on Haptics,* 4**,** 51-66.
COMAS, O., TAYLOR, Z. A., ALLARD, J., OURSELIN, S., COTIN, S. & PASSENGER, J. 2008. Efficient nonlinear FEM for soft tissue modelling and its GPU implementation within the open source framework SOFA. *Biomedical Simulation, Proceedings,* 5104**,** 28-39.
COSSERAT, E. & COSSERAT, F. 1909. *Théorie des corps déformables,* Paris,, A. Hermann et fils.
COTIN, S., DELINGETTE, H. & AYACHE, N. 1999. Real-time elastic deformations of soft tissues for surgery simulation. *Ieee Transactions on Visualization and Computer Graphics,* 5**,** 62-73.
COTTLE, R., PANG, J.-S. & STONE, R. E. 2009. *The linear complementarity problem,* Philadelphia, Society for Industrial and Applied Mathematics.
COUMANS, E. 2015. Bullet Physics Library.
COURTECUISSE, H. & ALLARD, J. 2009. Parallel Dense Gauss-Seidel Algorithm on Many-Core Processors. *Hpcc: 2009 11th Ieee International Conference on High Performance Computing and Communications***,** 139-147.
COURTECUISSE, H., COTIN, S., ALLARD, J. & SOLER, L. 2011. GPU-based interactive simulation of liver resection. *ACM SIGGRAPH 2011 Real-Time Live!* Vancouver, British Columbia, Canada: ACM.
CUETO, E. & CHINESTA, F. 2014. Real time simulation for computational surgery: a review. *Advanced Modeling and Simulation in Engineering Sciences,* 1**,** 1-18.
DARGAR, S., LAM, B., HORODYSKI, C., SANKARANARAYANAN, G. & DE, S. 2014. A Decoupled 2 DOF Force Feedback Mechanism for the Virtual Translumenal Endoscopic Surgical Trainer (VTEST. *Stud Health Technol Inform,* 196**,** 86-8.
DAWSON, S. L., COTIN, S., MEGLAN, D., SHAFFER, D. W. & FERRELL, M. A. 2000. Designing a computer-based simulator for interventional cardiology training. *Catheterization and Cardiovascular Interventions,* 51**,** 522-527.
DE MONTBRUN, S. L. & MACRAE, H. 2012. Simulation in surgical education. *Clin Colon Rectal Surg,* 25**,** 156-65.
DE VISSER, H., WATSON, M. O., SALVADO, O. & PASSENGER, J. D. 2011. Progress in virtual reality simulators for surgical training and certification. *Med J Aust,* 194**,** S38-40.
DELP, S. L., LOAN, J. P., HOY, M. G., ZAJAC, F. E., TOPP, E. L. & ROSEN, J. M. 1990. An interactive graphics-based model of the lower extremity to study orthopaedic surgical procedures. *IEEE Trans Biomed Eng,* 37**,** 757-67.
DEQUIDT, J., LENOIR, J. & COTIN, S. 2007. Interactive contacts resolution using smooth surface representation. *Medical Image Computing and Computer-Assisted Intervention- Miccai 2007, Pt 2, Proceedings,* 4792**,** 850-857.
DUNKIN, B., ADRALES, G. L., APELGREN, K. & MELLINGER, J. D. 2007. Surgical simulation: a current review. *Surgical Endoscopy and Other Interventional Techniques,* 21**,** 357-366.
DURATTI, L., WANG, F., SAMUR, E. & BLEULER, H. A Real-Time Simulator for Interventional Radiology. VRST '08 Proceedings of the 2008 ACM symposium on Virtual reality software and technology, 2008.
DURIEZ, C., COTIN, S., LENOIR, J. & NEUMANN, P. 2006. New approaches to catheter navigation for interventional radiology simulation. *Comput Aided Surg,* 11**,** 300-8.
EBERLY, D. H. 2010. *Game physics,* Burlington, MA, Morgan Kaufmann/Elsevier.
EPSTEIN, A. J., GROENEVELD, P. W., HARHAY, M. O., YANG, F. & POLSKY, D. 2013. Impact of minimally invasive surgery on medical spending and employee absenteeism. *JAMA Surg,* 148**,** 641-7.
ERICSON, C. 2005. *Real-time collision detection,* Amsterdam ; Boston, Elsevier.
ERLEBEN, K. 2007. Velocity-based shock propagation for multibody dynamics animation. *Acm Transactions on Graphics,* 26.
ERLEBEN, K. 2013. Numerical methods for linear complementarity problems in physics-based animation. *ACM SIGGRAPH 2013 Courses.* Anaheim, California: ACM.
FEATHERSTONE, R. 1987. *Robot dynamics algorithms,* Boston, Kluwer.
FRÂNCU, M. M. F. 2014. An Improved Jacobi Solver for Particle Simulation. *11th Workshop on Virtual Reality Interaction and Physical Simulation, VRIPHYS 2014.* Bremen.
FRISKEN-GIBSON, S. F. 3D ChainMail: a fast algorithm for reforming volumetric objects,. SIGGRAPH - Proceedings of the Symposium on Interactive 3D Graphics, 1997. 149—154.
GALLAGHER, A. G., RITTER, E. M., CHAMPION, H., HIGGINS, G., FRIED, M. P., MOSES, G., SMITH, C. D. & SATAVA, R. M. 2005. Virtual reality simulation for the operating room: proficiency-based training as a paradigm shift in surgical skills training. *Ann Surg,* 241**,** 364-72.

GALLAGHER, A. G. & SATAVA, R. M. 2015. Surgical Simulation: Seeing the Bigger Picture and Asking the Right Questions. *Annals of Surgery,* 262.

GALLAGHER, A. G. & TRAYNOR, O. 2008. Simulation in surgery: opportunity or threat? *Irish Journal of Medical Sciences***,** 177:283–287.

GILLEN, S., FIOLKA, A., KRANZFELDER, M., WOLF, P., FEITH, M., SCHNEIDER, A., MEINING, A., FRIESS, H. & FEUSSNER, H. 2011. Training of a standardized natural orifice transluminal endoscopic surgery cholecystectomy using an ex vivo training unit. *Endoscopy,* 43**,** 876-81.

GILLEN, S., WILHELM, D., MEINING, A., FIOLKA, A., DOUNDOULAKIS, E., SCHNEIDER, A., VON DELIUS, S., FRIESS, H. & FEUSSNER, H. 2009. The "ELITE" model: construct validation of a new training system for natural orifice transluminal endoscopic surgery (NOTES). *Endoscopy,* 41**,** 395-9.

GOLDSTEIN, H. 1981. Citation Classic - Classical Mechanics. *Current Contents/Engineering Technology & Applied Sciences***,** 16-16.

GOLUB, G. H. & VAN LOAN, C. F. 2013. *Matrix computations,* Baltimore, The Johns Hopkins University Press.

GOSS, V. G. A. 2003. *Snap buckling, writhing and loop formation in twisted rods.* University College London.

GOSWAMI, P., SCHLEGEL, P., SOLENTHALER, B. & PAJAROLA, R. 2010. Interactive SPH simulation and rendering on the GPU. *Proceedings of the 2010 ACM SIGGRAPH/Eurographics Symposium on Computer Animation.* Madrid, Spain: Eurographics Association.

GOULD, D. A., REEKERS, J. A., KESSEL, D. O., CHALMERS, N. C., SAPOVAL, M., PATEL, A. A., BECKER, G. J., LEE, M. J. & STOCKX, L. 2009. Simulation devices in interventional radiology: validation pending. *J Vasc Interv Radiol,* 20**,** S324-5.

GREGOIRE, M. & SCHOMER, E. 2007. Interactive simulation of one-dimensional flexible parts. *Computer-Aided Design,* 39**,** 694-707.

HADAP, S. 2006. Oriented strands: dynamics of stiff multi-body system. *Proceedings of the 2006 ACM SIGGRAPH/Eurographics symposium on Computer animation.* Vienna, Austria: Eurographics Association.

HALSTED, W. 1904. *The Training of the surgeon*.

HAMIT, F. 1993. *Virtual reality and the exploration of cyberspace,* Carmel, Ind., Sams Pub.

HWU, W.-M. 2012. *GPU computing gems,* Boston, MA, Morgan Kaufmann.

ISSENBERG, S. B., MCGAGHIE, W. C., HART, I. R., MAYER, J. W., FELNER, J. M., PETRUSA, E. R., WAUGH, R. A., BROWN, D. D., SAFFORD, R. R., GESSNER, I. H., GORDON, D. L. & EWY, G. A. 1999. Simulation technology for health care professional skills training and assessment. *JAMA,* 282**,** 861-6.

JAKOBSEN, T. 2001. Advanced character physics. *IN PROCEEDINGS OF THE GAME DEVELOPERS CONFERENCE 2001.*

JENSEN, U. J., JENSEN, J., OLIVECRONA, G. K., AHLBERG, G. & TORNVALL, P. 2013. Technical skills assessment in a coronary angiography simulator for construct validation. *Simul Healthc,* 8**,** 324-8.

JOHN, N. 2008. Design and implementation of medical training simulators. *Virtual Reality,* 12**,** 269-279.

KAHOL, K., SATAVA, R. M., FERRARA, J. & SMITH, M. L. 2009. Effect of short-term pretrial practice on surgical proficiency in simulated environments: a randomized trial of the "preoperative warm-up" effect. *J Am Coll Surg,* 208**,** 255-68.

KALAWSKY, R. S. 1993. *The science of virtual reality and virtual environments : a technical, scientific and engineering reference on virtual environments,* Wokingham, England ; Reading, Mass., Addison-Wesley.

KLEPPMANN, M. 2007. Simulation of colliding constrained rigid bodies.

KMOCH, P., BONANNI, U. & MAGNENAT-THALMANN, N. 2009. Hair simulation model for real-time environments. *Proceedings of the 2009 Computer Graphics International Conference.* Victoria, British Columbia, Canada: ACM.

KMOCH, P., BONANNI, U. & PELIK, J. 2010. Towards a GPU-only rod-based hair animation system. *ACM SIGGRAPH ASIA 2010 Posters.* Seoul, Republic of Korea: ACM.

KOCKRO, R. A., SERRA, L., TSENG-TSAI, Y., CHAN, C., YIH-YIAN, S., GIM-GUAN, C., LEE, E., HOE, L. Y., HERN, N. & NOWINSKI, W. L. 2000. Planning and simulation of neurosurgery in a virtual reality environment. *Neurosurgery,* 46**,** 118-35; discussion 135-7.

KOKKEVIS, E. 2004. Practical Physics for Articulated Characters. *Game Developers Conference.*

KROG, O. E. & ELSTER, A. C. 2012. Fast GPU-Based Fluid Simulations Using SPH. *Applied Parallel and Scientific Computing, Pt Ii,* 7134**,** 98-109.

KURODA, T. 2008. *VRASS(Virtual Reality Aided Surgical Simulation)* [Online]. Available: http://www.kuhp.kyoto-u.ac.jp/~mi/en/index.php?_research__med_vr.

LENOIR, J., COTIN, S., DURIEZ, C. & NEUMANN, P. 2006. Interactive physically-based simulation of catheter and guidewire. *Computers & Graphics-Uk,* 30**,** 416-422.

LI, S., QIN, J., GUO, J., CHUI, Y. P. & HENG, P. A. 2011. A Novel FEM-Based Numerical Solver for Interactive Catheter Simulation in Virtual Catheterization. *Int J Biomed Imaging,* 2011**,** 815246.

LIANG, Q., BORTHWICK, A. G. L. & STELLING, G. 2004. Simulation of dam- and dyke-break hydrodynamics on dynamically adaptive quadtree grids. *International Journal for Numerical Methods in Fluids,* 46**,** 127-162.

LIN, S., LEE, Y.-S. & NARAYAN, R. 2010. Heterogeneous Deformable Modeling of Bio-Tissues and Haptic Force Rendering for Bio-Object Modeling. *In:* NARAYAN, R., BOLAND, T. & LEE, Y.-S. (eds.) *Printed Biomaterials.* Springer New York.

LUBOZ, V., BLAZEWSKI, R., GOULD, D. & BELLO, F. 2009a. Real-time guidwire simulation in complex vascular models. *The Visual Computer,* 25**,** 827-834.

LUBOZ, V., HUGHES, C., GOULD, D., JOHN, N. & BELLO, F. 2009b. Real-time Seldinger technique simulation in complex vascular models. *Int J Comput Assist Radiol Surg,* 4**,** 589-96.

LUBOZ, V., ZHAI, J., ODETOYINBO, T., LITTLER, P., GOULD, D., HOW, T. & BELLO, F. 2011. Simulation of endovascular guidewire behaviour and experimental validation. *Comput Methods Biomech Biomed Engin,* 14**,** 515-20.

MACFADYEN, B. V., JR. & CUSCHIERI, A. 2005. Endoluminal surgery. *Surg Endosc,* 19**,** 1-3.

MACKLIN, M. & MULLER, M. 2013. Position Based Fluids. *Acm Transactions on Graphics,* 32.

MACKLIN, M. M., M.; CHENTANEZ, N.; KIM, TY.; 2014. Unified Particle Physics for Real-Time Applications. *ACM Transactions on Graphics (TOG),* 33**,** 104.

MARKS, S., WINDSOR, J. & WUNSCHE, B. 2007. Evaluation of Game Engines for Simulated Surgical Training. *Graphite 2007: 5th International Conference on Computer Graphics and Interactive Techniques in Australasia and Southern Asia, Proceedings***,** 273-+.

MARTIN, S., KAUFMANN, P., BOTSCH, M., GRINSPUN, E. & GROSS, M. 2010. Unified Simulation of Elastic Rods, Shells, and Solids. *Acm Transactions on Graphics,* 29.

MARTINEZ-MARTINEZ, F., RUPEREZ, M. J., MARTIN-GUERRERO, J. D., MONSERRAT, C., LAGO, M. A., PAREJA, E., BRUGGER, S. & LOPEZ-ANDUJAR, R. 2013. Estimation of the elastic parameters of human liver biomechanical models by means of medical images and evolutionary computation. *Computer Methods and Programs in Biomedicine,* 111**,** 537-549.

MEIER, U., LOPEZ, O., MONSERRAT, C., JUAN, M. C. & ALCANIZ, M. 2005. Real-time deformable models for surgery simulation: a survey. *Comput Methods Programs Biomed,* 77**,** 183-97.

MIRTICH, B. & CANNY, J. 1995. Impulse-based dynamic simulation. *Algorithmic Foundations of Robotics***,** 407-418.

MOGHUL, M. R., SODERGREN, M. H., CLARK, J., TEARE, J., YANG, G. Z. & DARZI, A. 2013. Education and Training in NOTES: A Systematic Review. *Surgical Innovation,* 20**,** 282-291.

MONTGOMERY, K., BRUYNS, C., BROWN, J., SORKIN, S., MAZZELLA, F., THONIER, G., TELLIER, A., LERMAN, B. & MENON, A. 2002. Spring: A general framework for collaborative, real-time surgical simulation. *Medicine Meets Virtual Reality 02/10,* 85**,** 296-303.

MOORE, P. & MOLLOY, D. 2007. A survey of computer-based deformable models. *Imvip 2007: International Machine Vision and Image Processing Conference, Proceedings***,** 55-64.

MORIS, D. N., BRAMIS, K. J., MANTONAKIS, E. I., PAPALAMPROS, E. L., PETROU, A. S. & PAPALAMPROS, A. E. 2012. Surgery via natural orifices in human beings: yesterday, today, tomorrow. *Am J Surg,* 204**,** 93-102.

MULLER, M., HEIDELBERGER, B., HENNIX, M. & RATCLIFF, J. 2007. Position based dynamics. *Journal of Visual Communication and Image Representation,* 18**,** 109-118.

MÜLLER, M. K., T.; CHENTANEZ, N.;. Fast Simulation of Inextensible Hair and Fur. VRIPHYS, 2012.

NEALEN, A., MULLER, M., KEISER, R., BOXERMAN, E. & CARLSON, M. 2006. Physically based deformable models in computer graphics. *Computer Graphics Forum,* 25**,** 809-836.

NEHME, J., SODERGREN, M. H., SUGDEN, C., AGGARWAL, R., GILLEN, S., FEUSSNER, H., YANG, G. Z. & DARZI, A. 2013. A randomized controlled trial evaluating endoscopic and laparoscopic training in skills transfer for novices performing a simulated NOTES task. *Surg Innov,* 20**,** 631-8.

NESME, M., FAURE, F. & PAYAN, Y. 2006. Hierarchical multi-resolution finite element model for soft body simulation. *Biomedical Simulation, Proceedings,* 4072**,** 40-47.

NOWINSKI, W. L. & CHUI, C. K. 2001. Simulation of interventional neuroradiology procedures. *International Workshop on Medical Imaging and Augmented Reality, Proceedings***,** 87-94.

NVIDIA. 2014. *CUDA C Programming Guide* [Online]. Available: http://docs.nvidia.com/cuda/cuda-c-programming-guide/ [Accessed 16/6/2014 2014].

PABST, S., KOCH, A. & STRASSER, W. 2010. Fast and Scalable CPU/GPU Collision Detection for Rigid and Deformable Surfaces. *Computer Graphics Forum,* 29**,** 1605-1612.

PAI, D. K. 2002. STRANDS: Interactive simulation of thin solids using cosserat models. *Computer Graphics Forum,* 21**,** 347-352.

PANG, W. M., QIN, J., CHUI, Y. P. & HENG, P. A. 2010. Fast Prototyping of Virtual Reality Based Surgical Simulators with PhysX-enabled GPU. *Transactions on Edutainment Iv,* 6250**,** 176-188.

PHARR, M. & FERNANDO, R. 2005. *GPU gems 2 : programming techniques for high-performance graphics and general-purpose computation,* Upper Saddle River, NJ, Addison-Wesley.

PHILLIPS, J., LADD, A. & KAVRAKI, L. E. 2002. Simulated knot tying. *2002 Ieee International Conference on Robotics and Automation, Vols I-Iv, Proceedings***,** 841-846.

PONSKY, J. L. 2006. Endoluminal surgery: past, present and future. *Surg Endosc,* 20 Suppl 2**,** S500-2.

PROVOT, X. 1995. Deformation constraints in a mass-spring model to describe rigid cloth behavior. *Graphics Interface '95 - Proceedings***,** 147-154.

PUNAK, S. & KURENOV, S. 2011a. Simplified Cosserat rod for interactive suture modeling. *Stud Health Technol Inform,* 163**,** 466-72.
PUNAK, S. & KURENOV, S. 2011b. A simulation framework for wound closure by suture for the endo stitch suturing instrument. *Stud Health Technol Inform,* 163**,** 461-5.
RATTNER, D. 2006. Introduction to NOTES White Paper. *Surg Endosc,* 20**,** 185.
RATTNER, D. W., HAWES, R., SCHWAITZBERG, S., KOCHMAN, M. & SWANSTROM, L. 2011. The Second SAGES/ASGE White Paper on natural orifice transluminal endoscopic surgery: 5 years of progress. *Surg Endosc,* 25**,** 2441-8.
RICHARDS, W. O. & RATTNER, D. W. 2005. Endoluminal and transluminal surgery: no longer if, but when. *Surg Endosc,* 19**,** 461-3.
ROSEN, K. R. 2008. The history of medical simulation. *J Crit Care,* 23**,** 157-66.
ROSENBLUM, R. E., CARLSON, W. E. & TRIPP, E. 1991. Simulating the structure and dynamics of human hair: Modelling, rendering and animation. *The Journal of Visualization and Computer Animation,* 2**,** 141-148.
RUSTICO, E., BILOTTA, G., HERAULT, A., DEL NEGRO, C. & GALLO, G. 2014. Advances in Multi-GPU Smoothed Particle Hydrodynamics Simulations. *Ieee Transactions on Parallel and Distributed Systems,* 25**,** 43-52.
RUTHENBECK, G. S. & REYNOLDS, K. J. 2013. Virtual reality surgical simulator software development tools. *Journal of Simulation,* 7**,** 101-108.
SALISBURY, K., CONTI, F. & BARBAGLI, F. 2004. Haptic rendering: Introductory concepts. *Ieee Computer Graphics and Applications,* 24**,** 24-32.
SANKARANARAYANAN, G., MATTHES, K., NEMANI, A., AHN, W., KATO, M., JONES, D. B., SCHWAITZBERG, S. & DE, S. 2013. Needs analysis for developing a virtual-reality NOTES simulator. *Surgical Endoscopy and Other Interventional Techniques,* 27**,** 1607-1616.
SATAVA, R. 2013. Virtual Reality: Current Uses in Medical Simulation and Future Opportunities & Medical Technologies that VR can Exploit in Education and Training. *Ieee Transactions on Visualization and Computer Graphics,* 19**,** Xii-Xii.
SATAVA, R. M. 1993. Virtual reality surgical simulator. The first steps. *Surg Endosc,* 7**,** 203-5.
SATAVA, R. M. 2001. Accomplishments and challenges of surgical simulation. *Surg Endosc,* 15**,** 232-41.
SATAVA, R. M. 2008. Historical review of surgical simulation--a personal perspective. *World J Surg,* 32**,** 141-8.
SELLE, A., LENTINE, M. & FEDKIW, R. 2008. A mass spring model for hair simulation. *Acm Transactions on Graphics,* 27.
SELZER, D. J. & DUNNINGTON, G. L. 2013. Surgical skills simulation: a shift in the conversation. *Ann Surg,* 257**,** 594-5.
SEYMOUR, N. E., GALLAGHER, A. G., ROMAN, S. A., O'BRIEN, M. K., BANSAL, V. K., ANDERSEN, D. K. & SATAVA, R. M. 2002. Virtual reality training improves operating room performance: results of a randomized, double-blinded study. *Ann Surg,* 236**,** 458-63; discussion 463-4.
SHUCAI, X. & WU-CHUN, F. Inter-block GPU communication via fast barrier synchronization. Parallel & Distributed Processing (IPDPS), 2010 IEEE International Symposium on, 19-23 April 2010 2010. 1-12.
SILCOWITZ-HANSEN, M., NIEBE, S. & ERLEBEN, K. 2010. A nonsmooth nonlinear conjugate gradient method for interactive contact force problems. *Visual Computer,* 26**,** 893-901.
SINGH, H., KALANI, M., ACOSTA-TORRES, S., EL AHMADIEH, T. Y., LOYA, J. & GANJU, A. 2013. History of simulation in medicine: from Resusci Annie to the Ann Myers Medical Center. *Neurosurgery,* 73 Suppl 1**,** 9-14.
SMITH, R. 2007. ODE: Open Dynamics Engine.
SPILLMANN, J. 2008. *CORDE: Cosserat Rod Elements for the Animation of Interacting Elastic Rods PhD Thesis.*
SPILLMANN, J. & HARDERS, M. 2010. Inextensible elastic rods with torsional friction based on Lagrange multipliers. *Computer Animation and Virtual Worlds,* 21**,** 561-572.
SPILLMANN, J. & TESCHNER, M. 2007. CORDE: Cosserat Rod Elements for the Dynamic Simulation of One-Dimensional Elastic Objects. *Symposium on Computer Animation 2007: Acm Siggraph/ Eurographics Symposium Proceedings***,** 63-72.
SPILLMANN, J. & TESCHNER, M. 2008. An adaptive contact model for the robust simulation of knots. *Computer Graphics Forum,* 27**,** 497-506.
SPIVAK, H. & HUNTER, J. G. 1997. Endoluminal surgery. *Surg Endosc,* 11**,** 321-5.
STAM, J. 2009. Nucleus: Towards a Unified Dynamics Solver for Computer Graphics. *2009 11th Ieee International Conference on Computer-Aided Design and Computer Graphics, Proceedings***,** 1-11.
SUTHERLAND, I. E. 1968. A head-mounted three dimensional display. *Proceedings of the December 9-11, 1968, fall joint computer conference, part I.* San Francisco, California: ACM.
TAN, J. & YANG, X. B. 2009. Physically-based fluid animation: A survey. *Science in China Series F-Information Sciences,* 52**,** 723-740.
TANG, W., LAGADEC, P., GOULD, D., WAN, T. R., ZHAI, J. H. & HOW, T. 2010. A realistic elastic rod model for real-time simulation of minimally invasive vascular interventions. *Visual Computer,* 26**,** 1157-1165.

TANG, W., TAO RUAN, W., GOULD, D. A., THIEN, H. & JOHN, N. W. 2012. A stable and real-time nonlinear elastic approach to simulating guidewire and catheter insertions based on Cosserat rod. *IEEE Trans Biomed Eng,* 59**,** 2211-8.

TERZOPOULOS, D., PLATT, J., BARR, A. & FLEISCHER, K. 1987. Elastically deformable models. *Proceedings of the 14th annual conference on Computer graphics and interactive techniques.* ACM.

THEETTEN, A., GRISONI, L., ANDRIOT, C. & BARSKY, B. 2008. Geometrically exact dynamic splines. *Computer-Aided Design,* 40**,** 35-48.

UMETANI, N., SCHMIDT, R. & STAM, J. 2014. Position-based elastic rods. *ACM SIGGRAPH 2014 Talks.* Vancouver, Canada: ACM.

VAN DER HEIJDEN, G. H. M., NEUKIRCH, S., GOSS, V. G. A. & THOMPSON, J. M. T. 2003. Instability and self-contact phenomena in the writhing of clamped rods. *International Journal of Mechanical Sciences,* 45**,** 161-196.

VERLET, L. 1967. Computer "Experiments" on Classical Fluids. I. Thermodynamical Properties of Lennard-Jones Molecules. *Physical Review,* 159**,** 98-103.

VIDAL, F. P., BELLO, F., BRODLIE, K. W., JOHN, N. W., GOULD, D., PHILLIPS, R., AVIS, N. J. & HIRSCH, T. 2006. Principles and applications of computer graphics in medicine. *Computer Graphics Forum,* 25**,** 113-137.

VRS. 2015. *Virtual Reality* [Online]. Available: http://www.vrs.org.uk/ [Accessed 20.02.2015.

WANG, F., BURDET, E., DHANIK, A., POSTON, T. & TEO, C. L. 2005a. Dynamic thread for real-time knot-tying. *World Haptics Conference: First Joint Eurohaptics Conference and Symposium on Haptic Interfaces for Virutual Environment and Teleoperator Systems, Proceedings***,** 507-508.

WANG, F., BURDET, E., VUILLEMIN, R. & BLEULER, H. 2005b. Knot-tying with visual and force feedback for VR laparoscopic training. *2005 27th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, Vols 1-7***,** 5778-5781.

WANG, F., DURATTI, L., SAMUR, E., SPAELTER, U. & BLEULER, H. 2007. A computer-based real-time simulation of interventional radiology. *Conf Proc IEEE Eng Med Biol Soc,* 2007**,** 1742-5.

WANG, Y. 2009. *Simulation of Blood Flow and Contrast Medium Propagation - MSc thesis.* MSc, Imperial College London.

WANG, Z. W., LI, S. A., YANG, L. P. & HAO, A. M. 2012. Real-time CUDA based collision detection and Physics based collision response simulation. *2012 International Symposium on Information Science and Engineering (Isise)***,** 250-254.

WARD, K., BERTAILS, F., KIM, T. Y., MARSCHNER, S. R., CANI, M. P. & LIN, M. C. 2007. A survey on hair modeling: Styling, simulation, and rendering. *Ieee Transactions on Visualization and Computer Graphics,* 13**,** 213-234.

WHO, W. H. O. 2011. Global status report on noncommunicable diseases Geneva.

WICKHAM, J. E. 1987. *The new surgery*.

WILSON, M. S., MIDDLEBROOK, A., SUTTON, C., STONE, R. & MCCLOY, R. F. 1997. MIST VR: a virtual reality trainer for laparoscopic surgery assesses performance. *Ann R Coll Surg Engl,* 79**,** 403-4.

WONG, T. H., LEACH, G. & ZAMBETTA, F. 2012. Virtual subdivision for GPU based collision detection of deformable objects using a uniform grid. *Visual Computer,* 28**,** 829-838.

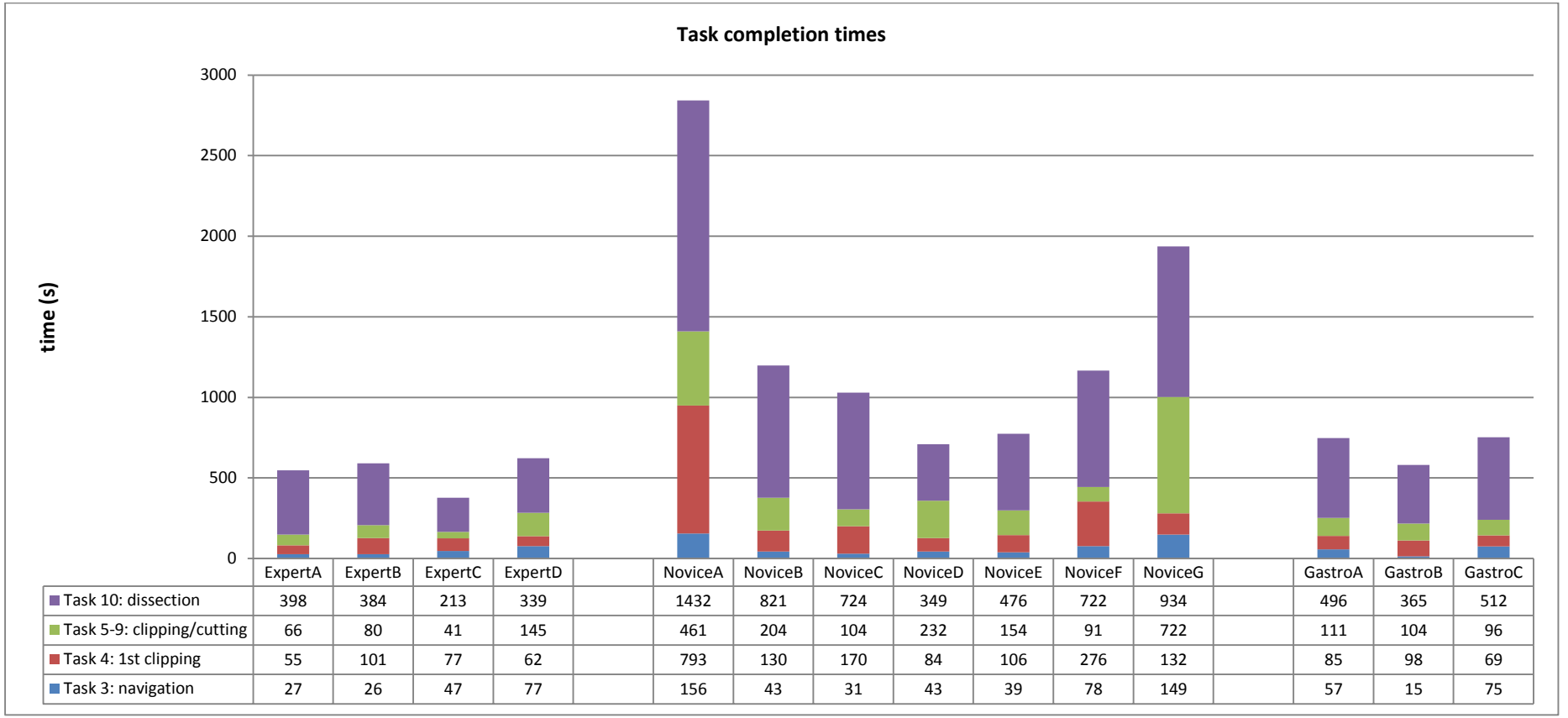
WONG, T. H., LEACH, G. & ZAMBETTA, F. 2014. An adaptive octree grid for GPU-based collision detection of deformable objects. *Visual Computer,* 30**,** 729-738.

YOUNGBLOOD, P. L., SRIVASTAVA, S., CURET, M., HEINRICHS, W. L., DEV, P. & WREN, S. M. 2005. Comparison of training on two laparoscopic simulators and assessment of skills transfer to surgical performance. *J Am Coll Surg,* 200**,** 546-51.

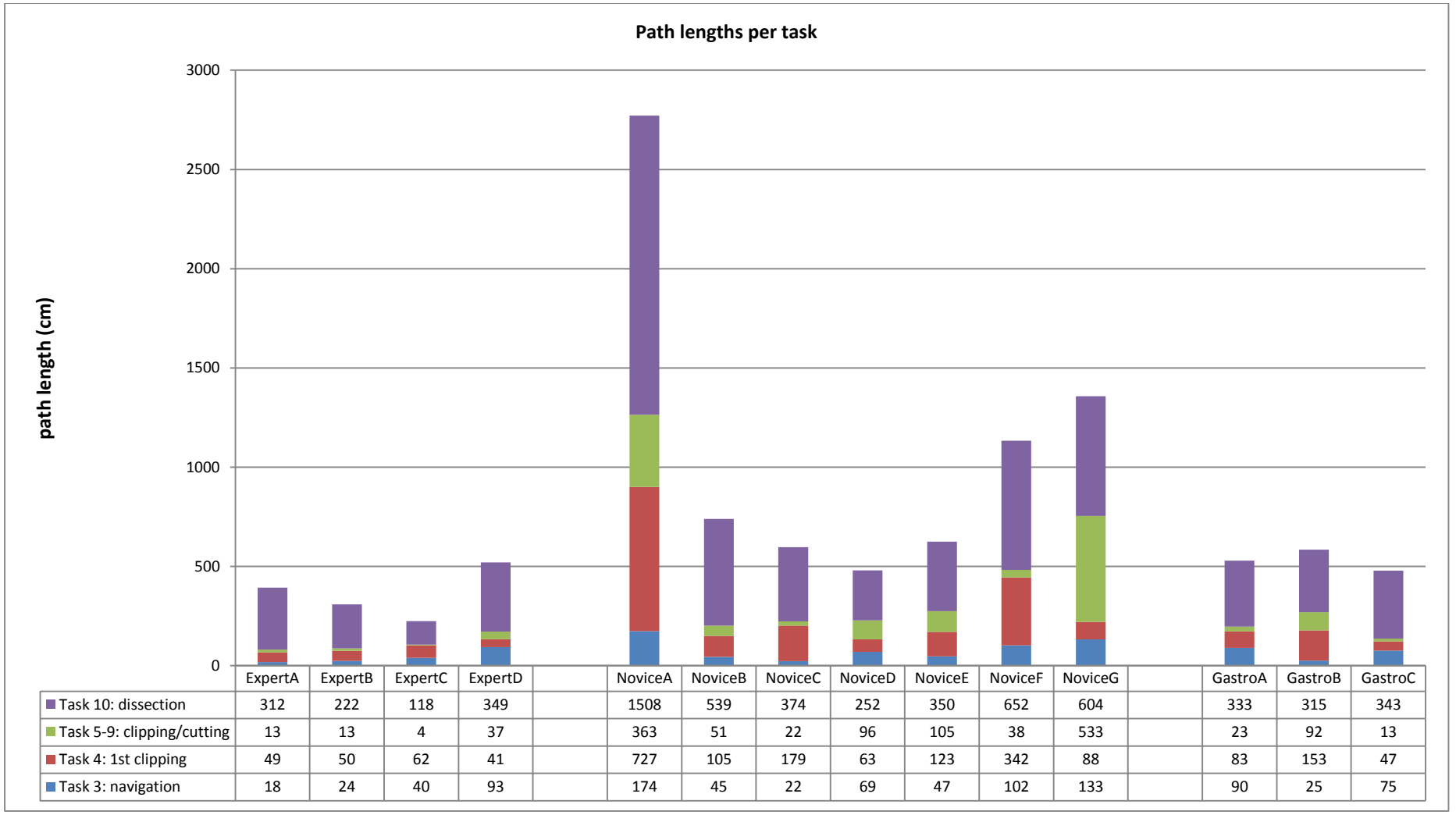
ZENDEJAS, B., BRYDGES, R., HAMSTRA, S. J. & COOK, D. A. 2013. State of the evidence on simulation-based training for laparoscopic surgery: a systematic review. *Ann Surg,* 257**,** 586-93.

ZIENKIEWICZ, O. C. & TAYLOR, R. L. 2000. *The finite element method,* Oxford ; Boston, Butterworth-Heinemann.

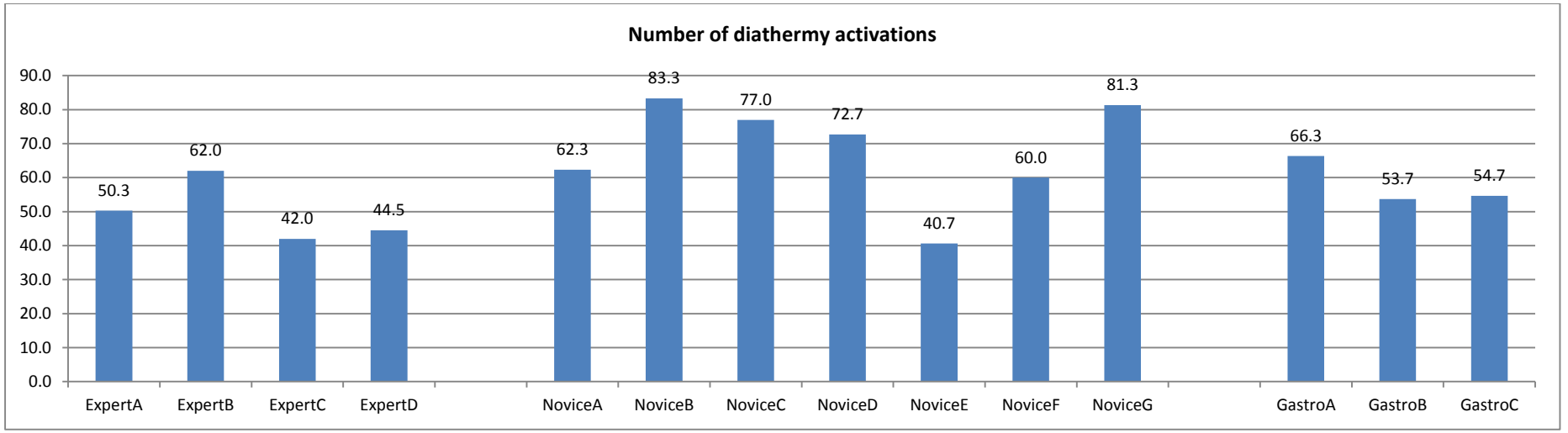
APPENDIX A: NOTES VALIDATION DETAILED CHARTS

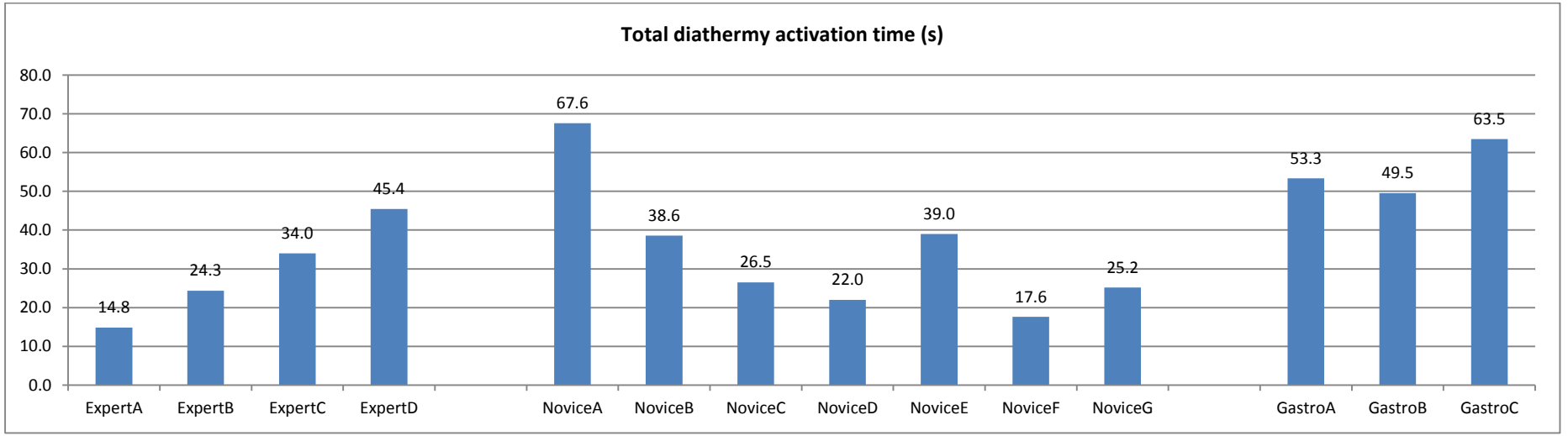


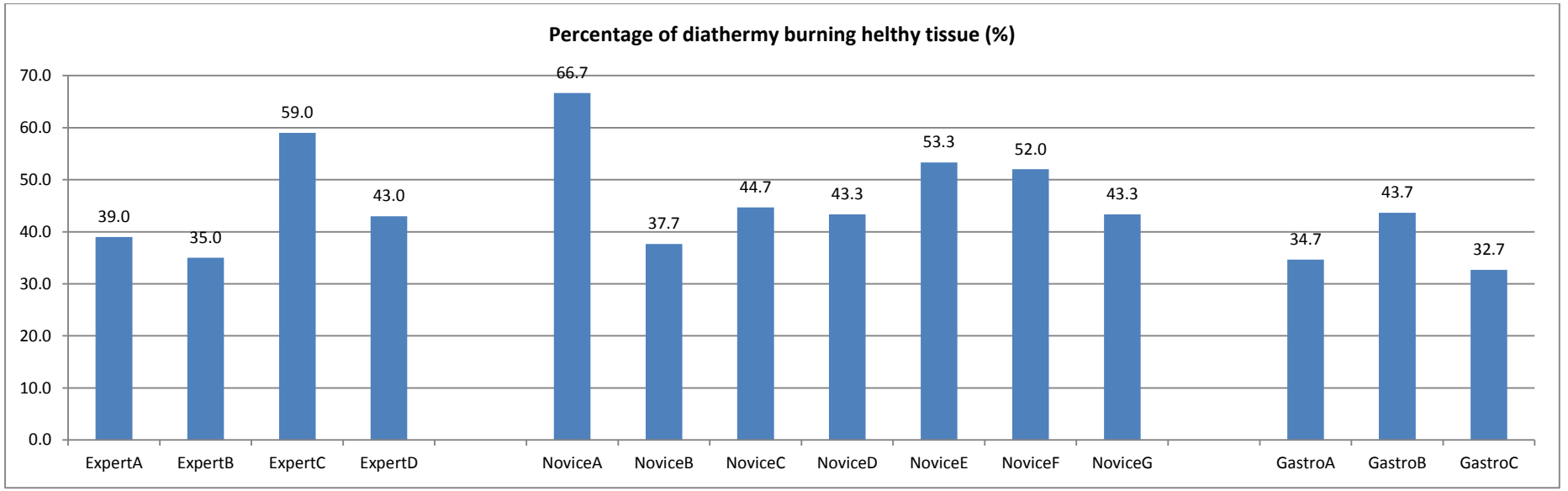
	ExpertA	ExpertB	ExpertC	ExpertD		NoviceA	NoviceB	NoviceC	NoviceD	NoviceE	NoviceF	NoviceG		GastroA	GastroB	GastroC
Task 10: dissection	398	384	213	339		1432	821	724	349	476	722	934		496	365	512
Task 5-9: clipping/cutting	66	80	41	145		461	204	104	232	154	91	722		111	104	96
Task 4: 1st clipping	55	101	77	62		793	130	170	84	106	276	132		85	98	69
Task 3: navigation	27	26	47	77		156	43	31	43	39	78	149		57	15	75

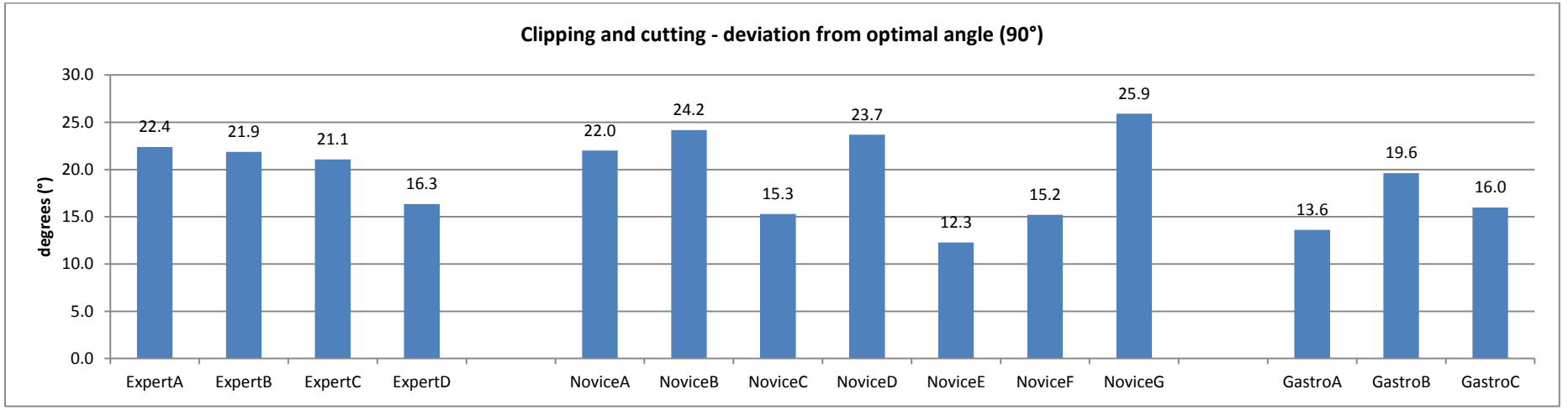


	ExpertA	ExpertB	ExpertC	ExpertD		NoviceA	NoviceB	NoviceC	NoviceD	NoviceE	NoviceF	NoviceG		GastroA	GastroB	GastroC
Task 10: dissection	312	222	118	349		1508	539	374	252	350	652	604		333	315	343
Task 5-9: clipping/cutting	13	13	4	37		363	51	22	96	105	38	533		23	92	13
Task 4: 1st clipping	49	50	62	41		727	105	179	63	123	342	88		83	153	47
Task 3: navigation	18	24	40	93		174	45	22	69	47	102	133		90	25	75


Number of diathermy activations
90.0
80.0
70.0
60.0
50.0
40.0
30.0
20.0
10.0
0.0
50.3
62.0
42.0
44.5
62.3
83.3
77.0
72.7
40.7
60.0
81.3
66.3
53.7
54.7
ExpertA
ExpertB
ExpertC
ExpertD
NoviceA
NoviceB
NoviceC
NoviceD
NoviceE
NoviceF
NoviceG
GastroA
GastroB
GastroC


Total diathermy activation time (s)
80.0
70.0
60.0
50.0
40.0
30.0
20.0
10.0
0.0
14.8
24.3
34.0
45.4
67.6
38.6
26.5
22.0
39.0
17.6
25.2
53.3
49.5
63.5
ExpertA
ExpertB
ExpertC
ExpertD
NoviceA
NoviceB
NoviceC
NoviceD
NoviceE
NoviceF
NoviceG
GastroA
GastroB
GastroC

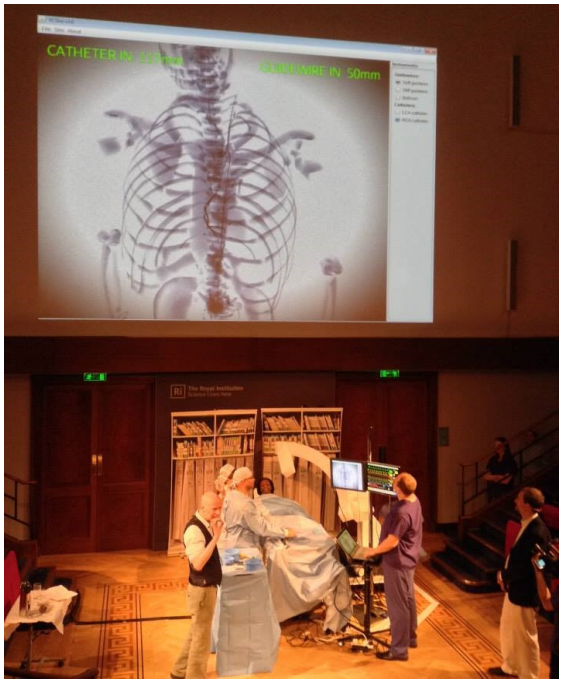

Percentage of diathermy burning helthy tissue (%)
70.0
60.0
50.0
40.0
30.0
20.0
10.0
0.0
39.0
35.0
59.0
43.0
66.7
37.7
44.7
43.3
53.3
52.0
43.3
34.7
43.7
32.7
ExpertA
ExpertB
ExpertC
ExpertD
NoviceA
NoviceB
NoviceC
NoviceD
NoviceE
NoviceF
NoviceG
GastroA
GastroB
GastroC

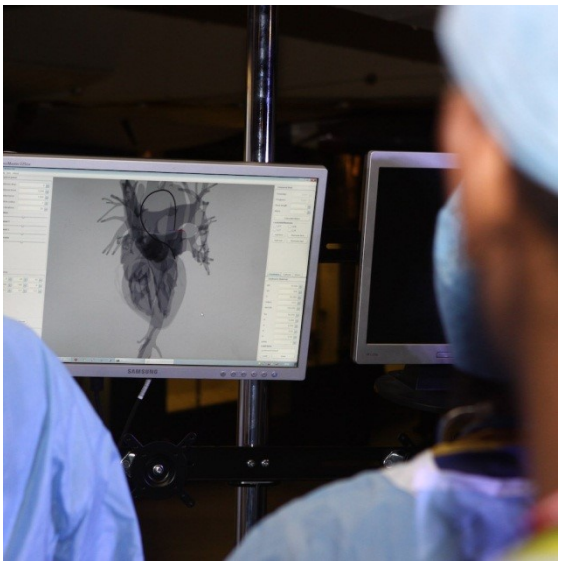

Clipping and cutting - deviation from optimal angle (90°)
degrees (°)
30.0
25.0
20.0
15.0
10.0
5.0
0.0
22.4
21.9
21.1
16.3
22.0
24.2
15.3
23.7
12.3
15.2
25.9
13.6
19.6
16.0
ExpertA
ExpertB
ExpertC
ExpertD
NoviceA
NoviceB
NoviceC
NoviceD
NoviceE
NoviceF
NoviceG
GastroA
GastroB
GastroC

APPENDIX B: PUBLIC ENGAGEMENT AND SCIENCE COMMUNICATION

Both simulators were demonstrated at a number of public engagement and science communications events across the UK and Europe:

- ✓ The Royal Institution Lates 2015
- ✓ The Science Picnic 2014 (Warsaw, Poland, 100.000+ visitors)
- ✓ The Times Cheltenham Science Festival 2014 (45.000 visitors)
- ✓ The Big Bang Fair 2013 (ExCeL, London, 65.000 visitors)
- ✓ The Big Bang Fair 2012 (NEC, Birmingham, 49.000 visitors)
- ✓ "Teaching your eyes to feel" - The Royal Institution of Great Britain 2014
- ✓ "Teaching your fingers to see" - The Royal Institution of Great Britain 2013
- ✓ Science Museum Lates
- ✓ Natural History Museum
- ✓ Imperial Festival
- ✓ Imperial West Launch Event


CATHETER IN: 111mm
GUIDEWIRE IN: 50mm




C21
IMPERIAL COLLEGE LONDON

APPENDIX C: NOVISE VALIDATION DOCUMENTATION

Validation of NOTES-SIM

A novel VR simulator for NOTES[1]

Scientific Protocol

July 2013

[1] Natural Orifice Transluminal Endoscopic Surgery

1. RATIONALE AND AIMS OF THE STUDY

Natural Orifice Transluminal Endoscopic Surgery (NOTES) is a surgical technique in which operations are performed by passing an endoscope through a natural orifice such as the mouth, anus, urethra and vagina then through an internal incision to reach the operative site. It has the potential to bring about a paradigm shift in surgery by offering the benefits of established minimally invasive techniques such as laparoscopy (reduced post-operative pain, morbidity and hospital stay) with scar-free surgery.

One of the main challenges in implementing NOTES into mainstream clinical practice is how to safely train operators in its application, particularly as NOTES requires both endoscopic and surgical skills; hitherto chiefly acquired by gastroenterologists and surgeons in isolation.

Virtual reality simulation has been shown to be a safe and effective training tool in both endoscopy and minimally invasive surgery. We have therefore developed a novel NOTES virtual reality simulator, which allows operators to develop skills in a safe environment and provides them with objective feedback on their performance.

The aim of this study is to establish face, content and construct validity for our 'NOTES-SIM' simulator in performing a transgastric hybrid NOTES cholecystectomy.

2. METHODS

Materials

The experimental apparatus consists of a) A physical, force feedback human-computer-interface (the haptic device). b) A real-time software simulation (the simulation); The haptic device consists of an enclosed black box, dimensions approximately 550x260x180mm, into which passes a length of hose 15mm diameter through a small circular opening (Appendix 1 Fig. 1). The hose can be pushed or pulled through the opening (total travel 220mm) and rotated freely. The angle and insertion distance are measured and read by the simulation. DC motors connected to the hose inside the box provide both linear and rotational forces and are commanded by the simulation. At the end of the hose (approximately 1.5m) a plastic replica of a standard endoscopic hand piece is attached (Appendix 1 Fig. 2). The hand piece consists of: 2 thumb wheels, 2 push buttons and 2 thin wires representing the endoscopic tool wires. The angle of the thumb wheels, button pressed states and insertion distance of the tool wires are all measured electronically and read into the simulation. Additionally, a double foot pedal (Appendix 1 Fig. 3) is placed on the floor and is freely positionable by the user. The state (on/off) of both pedals is also read in and used to control functions in the simulation.

The simulation software runs a 2kHz update loop (2000 updates per 1 second of simulation). During each iteration:

- receives movements from the haptic device
- calculates the motion of the virtual flexible endoscope
- processes the interactions of the endoscope with internal organs
- measures and stores the performance metrics
- calculates and sends the force feedback back to the haptic device

A parallel 60Hz loop renders a 3D visual output of the endoscope camera**.**

TASKS

The procedure starts with the endoscope partially inserted into the oesophagus. It is divided into three main tasks, each with several subtasks:

Navigation (Appendix 2 Fig. 1.a – 1.c)

- navigate to the first checkpoint (red glowing sphere) at the distal oesophagus
- navigate to the second checkpoint (red glowing sphere) inside the stomach
- go through the red glowing torus from the stomach to the peritoneal cavity

Clipping and cutting (Appendix 2 Fig. 2.a – 2.d)

- use the clipper (right tool) to clip the the cystic artery at two prescribed points
- use the clipper (right tool) to clip the the cystic duct at two prescribed points
- use the cutter (left tool) in between clips to cut the artery
- use the cutter (left tool) in between clips to cut the cystic duct

Gallbladder dissection (Appendix 2 Fig. 3.a – 3.b)

- use the diathermy tool (right tool) to dissect the connective tissue attaching the gallbladder to the liver bed (use +\- keys to increase / decrease gallbladder retraction)
- use the grasper (left tool) to grab the gallbladder and retrieve it via the stomach

When each (sub) task is completed, the simulation will automatically advance to the next one and select the appropriate instruments.

The length of the hose is less than that of a real endoscope. If the length of available hose is exceeded, then the screen will fade out, the simulation will be paused, and the user will be asked to reset the insertion of the hose. This is a necessary trade-off to keep the whole simulator compact and portable.

METRICS

The software constantly tracks and stores all the movements of the haptic device and of the virtual endoscope. The software also stores the following performance metrics:

For all tasks:

- task completion time
- path length of the tip traversed during the task
- max and average force received by the endoscope's tip during the task
- max and average force received by the endoscope's tip section (last 5cm) during the task

For the clipping and cutting subtasks:

- clipping /cutting distance in centimeters from the indicated point (centre of glowing sphere)
- clipping /cutting angle between the clipping / cutting tool and the surface of the duct (optimal = 90 degrees)
- number of clippings /cuttings
- degree of instrument protrusion during the operation (one should avoid protruding the instruments from the tip of the endoscope when not in use to avoid unintentional damage to tissues)

For the gallbladder dissection subtasks:

- number of instances diathermy is activated
- total time diathermy is activated
- percentage of time burning non-target tissue

Participants

Experts: We have defined experts as surgeons who have performed 10 or more animal-model or human NOTES procedures independently. We aim to recruit between 5 and 10 experts. This is analogous to previous studies of a similar design [1-2].

Novices: We have defined novices as surgeons who have performed fewer than 10 computer simulated, animal-model or human NOTES procedures independently. In addition, in order to prevent construct validation of the simulator as a tool for acquiring endoscopic (as opposed to NOTES) skills, we have stipulated that novices must have performed at least 10 endoscopic procedures independently on patients. Equally, in order to prevent construct validation of the simulator as a tool

for acquiring surgical skills, we have stipulated that novices must have performed at least 5 laparoscopic cholecystectomy procedures independently on patients. We aim to recruit between 5 and 10 novices. This is analogous to previous studies of a similar design.

Study Design

Participant data: Participants' operative experience, experience of videogames, demographic data, and interest in virtual reality simulation will be recorded with an online questionnaire (www.surveymonkey.com, Appendix 3). **In addition, we intend to also video record the procedure (operative screen and scope manipulation).**

Simulation: All participants will be required to complete 3 transgastric hybrid (ie. with gallbladder retraction done by an assisting surgeon with a laparoscopic grasper) NOTES cholecystectomies in line with how clinical hybrid NOTES is being performed currently. The aforementioned operative metrics will be recorded. The participants' performance, as determined by these metrics, will be analyzed in relation to the 3 operative tasks as well as the procedure as a whole and an average taken. Prior to performing their first procedure, all participants will be given a technical instruction sheet outlining the nature of the simulation. The aim of this sheet is to give a brief overview of the equipment, tasks and factors, which will differ to real life owing to the limitations of performing the procedure in a simulated setting. Participants will also be informed of what help they may receive from the researcher (who will be acting as assisting surgeon) during the procedure; namely holding the endoscope in a particular position, activating the instruments and retracting the gallbladder. These actions would be performed by an assisting surgeon in real life. In order to prevent bias the researcher will only act following a direct instruction from the operator. The instruction sheet is designed so as not to instruct on the particular challenges of performing a cholecystectomy with NOTES in order that we may better detect any potential differences in performance between novices and experts. After reading the instruction sheet, participants will be given a maximum of 3 minutes to familiarize themselves with basic navigation of the endoscope and how to operate the instruments. This will be done on a non-anatomical simulated module (Appendix 2 Fig. 4) and no metrics will be recorded. Prior to commencing their first recorded procedure, participants will be given the opportunity to ask questions relating to the practicalities of the simulation, but will not be allowed to request any technical advice as to how best to perform the procedure. No time limit will be set for the 3 recorded procedures.

Face validity: Face validity will be evaluated by asking experts to complete the aforementioned questionnaire after completing their 3 procedures. The questionnaire will assess graphical appearance, behavior of tools and tissue, difficulty of the procedure and overall realism.

Content validity: Content validity will be evaluated by asking experts to complete the aforementioned questionnaire after completing their 3 procedures. The questionnaire will assess the adequacy of the simulated tasks and perceived utility of the simulator as a training tool for NOTES.

Construct validity: Construct validity will be evaluated by comparing operative performance metrics of novice and expert subjects.

Data collection:

Questionnaire responses will be stored online in a secure online account (www.surveymonkey.com). Operative metrics and video recordings will be stored electronically on a secure departmental hard drive. The participants will be assigned a code on all data forms to ensure their anonymity. The document containing their real names will be kept in a separate secure electronic file.

Data Analysis

All data will remain anonymous. Data will be analyzed using a statistics package (Statistical Package for the Social Sciences (SPSS) version 18.0.1). Descriptive statistics and frequencies will be calculated with appropriate methods according to the type of data. Significance will be set at $p<0.05$.

Site

The project will take place in the Patterson Centre of St Mary's Hospital, Paddington, London, W2 1BL. We may also recruit participants attending the Hamlyn Symposium.

Ethics

Institutional review board approval will be obtained from the Imperial College Research and Ethics Committee.

Funding

The project has been funded by the EPSRC (Engineering and Physical Sciences Research Council).

3. INTENDED OUTCOME

It is hoped that the study will lead to the face, content and construct validation of the NOTES-SIM simulator, such that it may be used for the safe and effective training of NOTES.

REFERENCES

[1] Bittner JG 4th, Mellinger JD, Imam T, Schade RR, Macfadyen BV Jr. Face and construct validity of a computer-based virtual reality simulator for ERCP. *Gastrointest Endosc.* 2010 Feb;71(2):357-64. doi: 10.1016/j.gie.2009.08.033.

[2] Jensen UJ, Jensen J, Olivecrona GK, Ahlberg G, Tornvall P. Technical Skills Assessment in a Coronary Angiography Simulator for Construct Validation. *Simul Healthc.* 2013 Apr 17. [Epub ahead of print]

Appendix 1

Fig. 1 NOTES-SIM Hardware

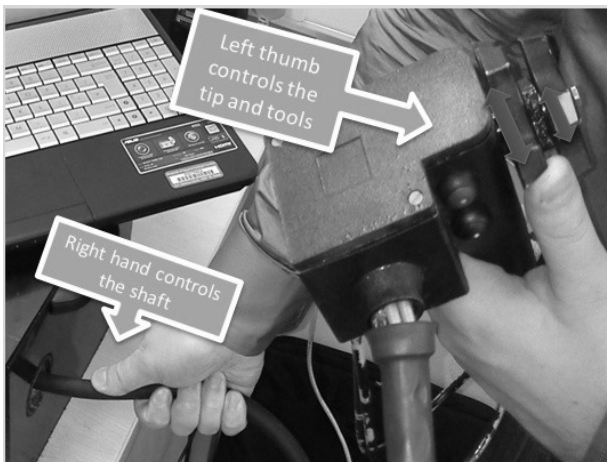


Fig. 2 NOTES-SIM Hand Piece

Fig. 3 NOTES-SIM Foot Pedals

Appendix 2

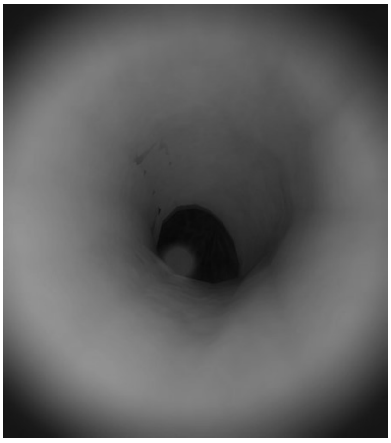


Fig. 1.a Navigating to the first checkpoint at the distal end of the oesophagus

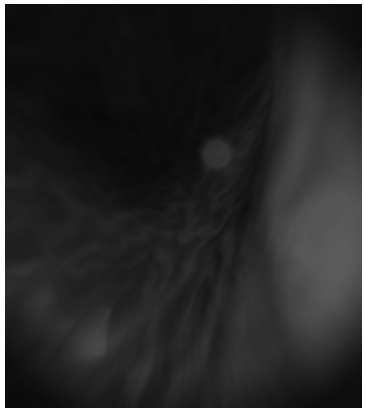


Fig. 1.b Navigating to the second checkpoint inside the stomach

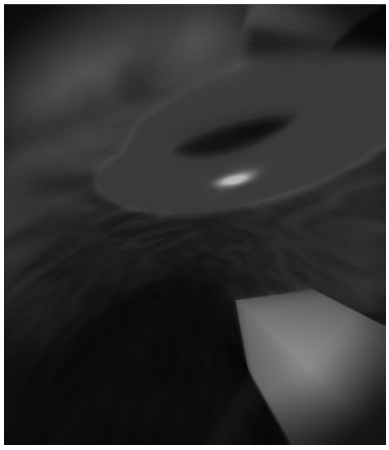


Fig. 1.c Navigating through the red glowing torus into the peritoneal cavity

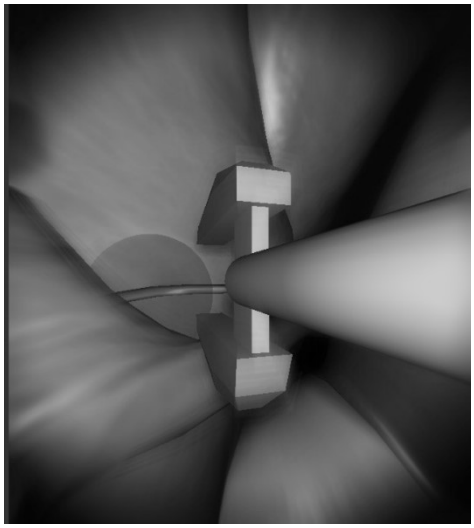


Fig. 2.a Clipping the cystic artery

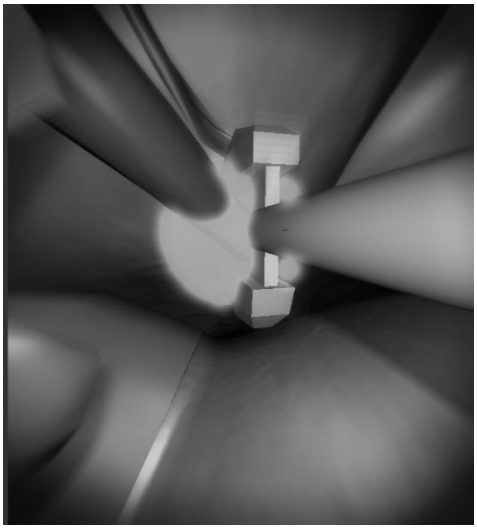


Fig. 2.b Clipping the cystic duct

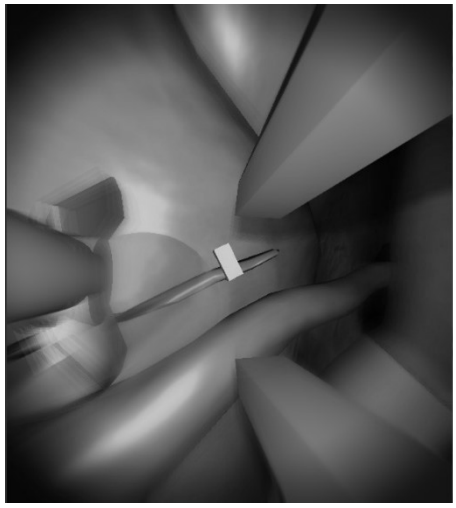


Fig. 2.c Cutting the cystic artery

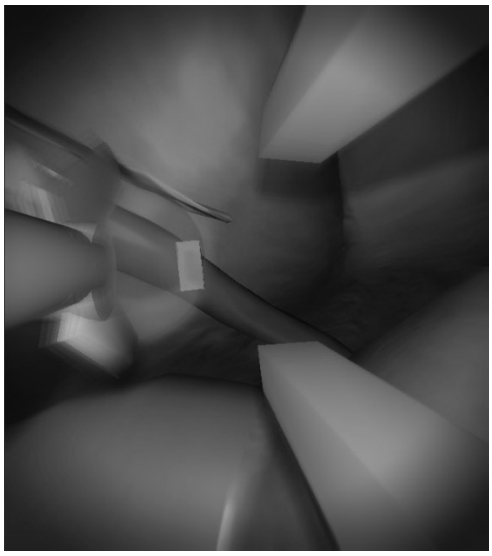


Fig. 2.d Cutting the cystic duct

Fig. 3.a Dissecting the gallbladder off the liver with the diathermy

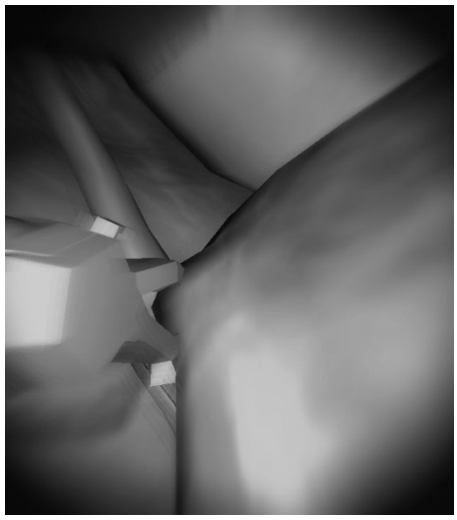


Fig. 3.b Removing the gallbladder with the grasper

Appendix 3

Online questionnaire accessed via the following link:

https://www.surveymonkey.com/s/NOTES_QUESTIONNAIRE

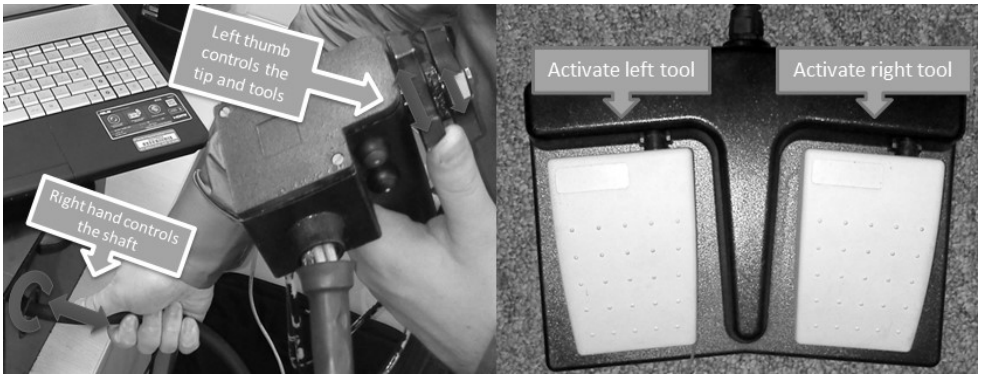
Validation of NOTES-SIM

A novel virtual reality simulator for NOTES[2]

Technical Instruction Sheet

As part of this study you have been requested to perform 3 simulated transgastric hybrid NOTES cholecystectomies on a virtual reality simulator. This sheet has been designed to provide you with an introduction to the equipment that you will be using, as well as an overview of the nature of the simulation.

EQUIPMENT



- The above picture shows the endoscope that you will be using.
- Your left hand activates the controls:
  - The large wheel is the up / down angulation control.
  - The small wheel is the left / right angulation control.
  - *N.B You will not be required to insufflate / deflate nor to apply water*.
- Your right hand:
  - controls the endoscope shaft:
    - Insert / withdraw the shaft further into the machine to advance / withdraw the tip of the endoscope. *N.B. The length of the shaft is less than that of a real endoscope. During the simulation, you may occasionally exceed the available length of the shaft. If this happens, the screen will fade out, the simulation will be paused and you will be asked to withdraw the shaft. Once the shaft has been withdrawn to an appropriate position, the simulation will re-commence with the tip of your endoscope in an unaltered position on the screen*

[2] Natural Orifice Transluminal Endoscopic Surgery

- Twist the shaft clockwise / anticlockwise to torque the endoscope.
- **You may ask your assistant (who will be one of the study investigators) to hold the shaft in a given position at any time.**

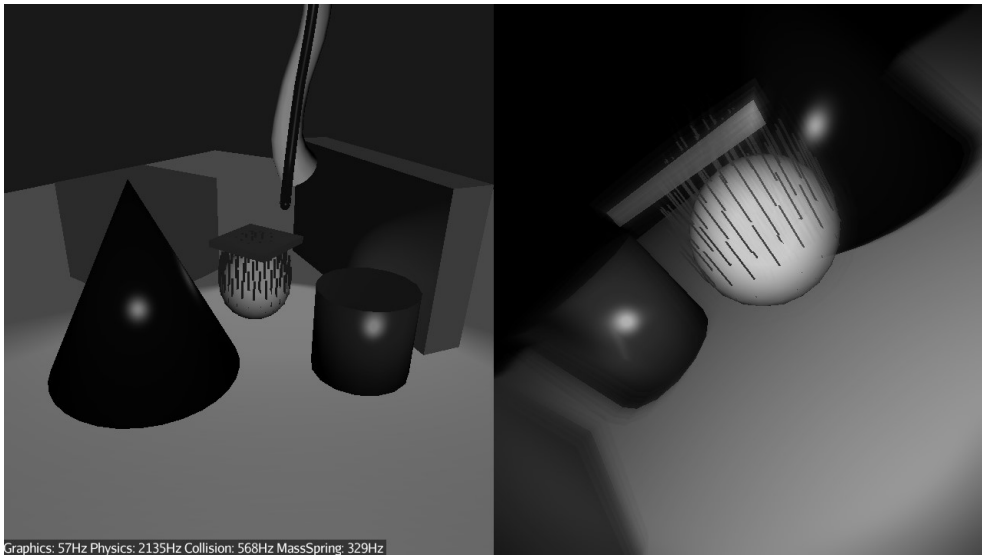
- advances / withdraws the endoscope instrument wires:
  - Advance / withdraw the endoscope instrument wires down the endoscope to advance / withdraw the instruments.
  - You may activate the tools with the foot pedal.
  - You may ask your assistant to advance / withdraw the instruments for you.

QUESTIONS

You may now ask the investigator any questions you have. Please note that although the investigator can answer questions relating to the practicalities of the simulation they cannot offer any technical advice as to how best to perform the procedure.

BASIC SIMULATION

Prior to commencing the procedural simulation with the transgastric hybrid NOTES cholecystectomy, you will be given a maximum of 3 minutes to practice basic endoscopic navigation in a test environment. Your performance during this basic simulation will not be analysed.

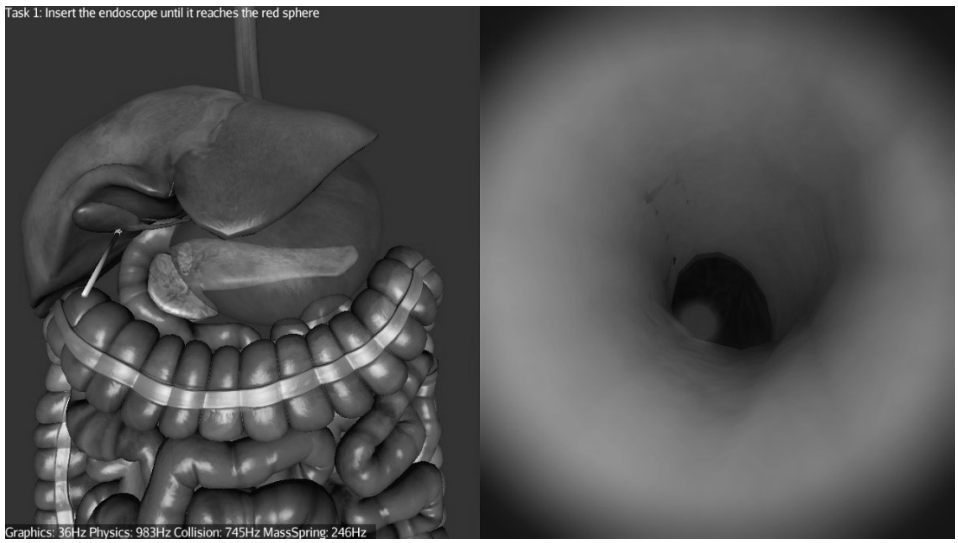


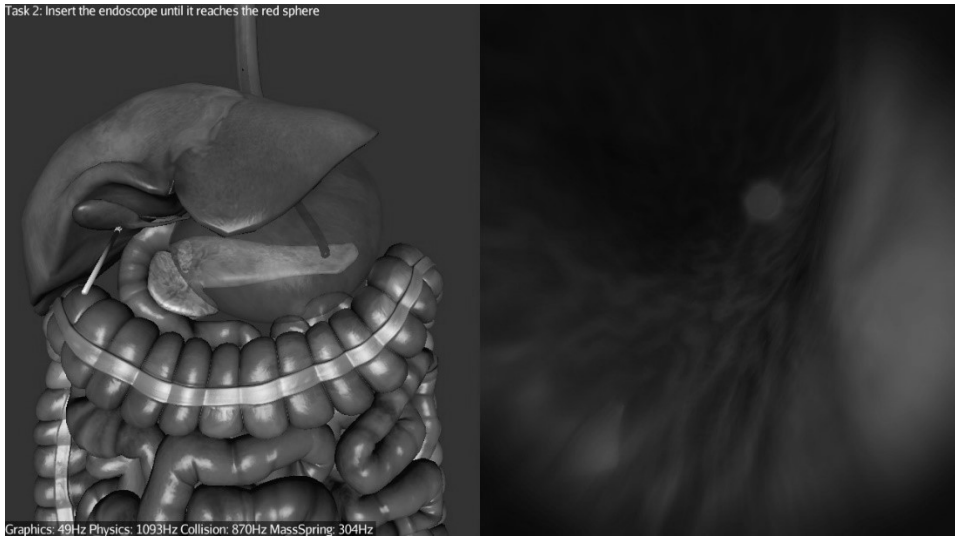
PROCEDURAL SIMULATION

The procedural transgastric hybrid NOTES cholecystectomy simulation will start with the endoscope in the oesophagus. You will be required to complete the below steps as part of the operation. There

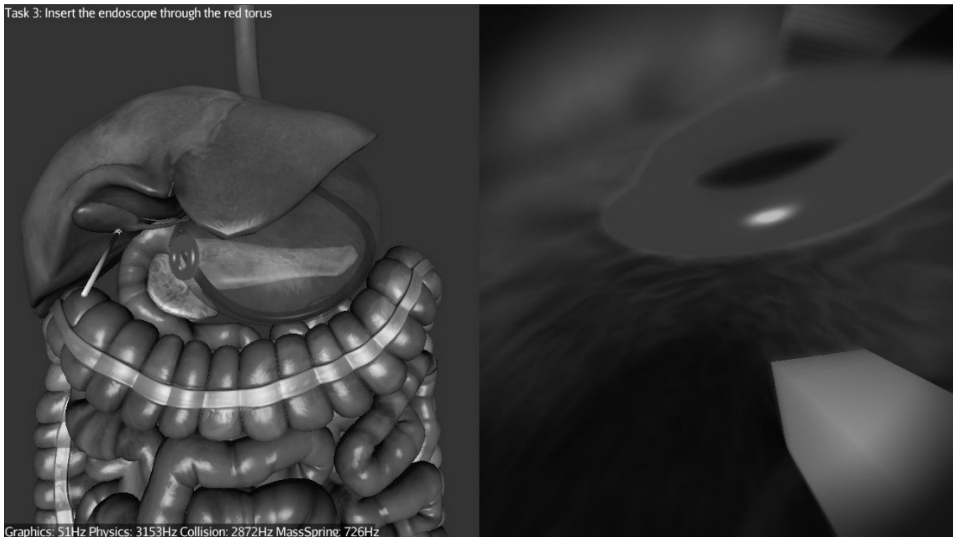
is no time limit. You should take the same care as you would if you were operating on a human. *N.B For the purposes of the simulation, the instruments necessary for each part of the procedure will become available to you automatically at the appropriate stage.*

1. Navigate to the checkpoints (red glowing spheres

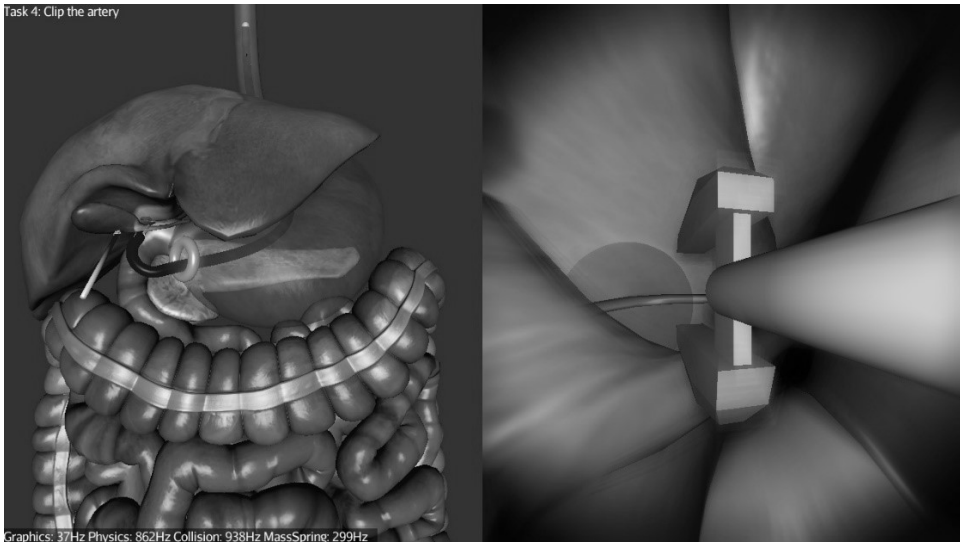


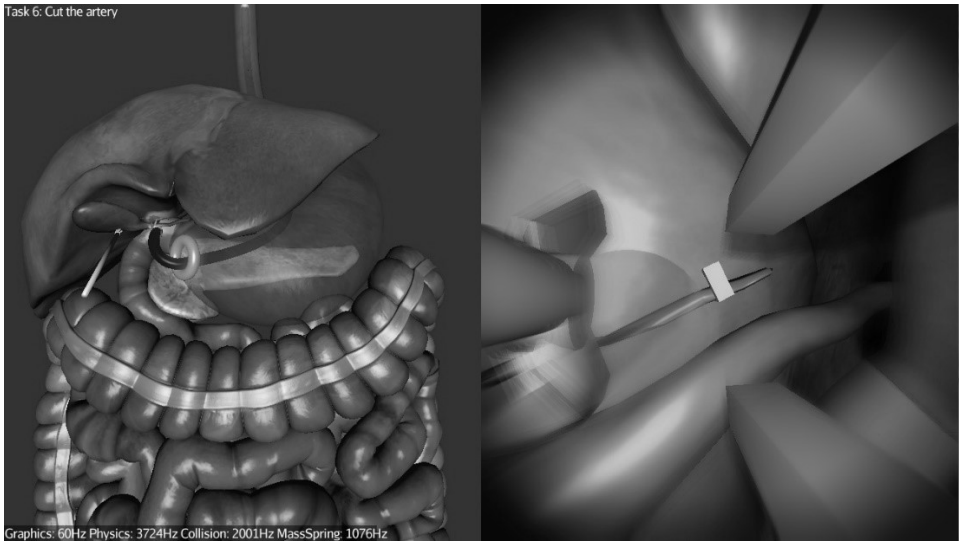


2. Navigate through the red glowing torus from the stomach and into the abdominal cavity. *N.B For the purposes of the simulation you will not be required to dissect a hole in the stomach.*

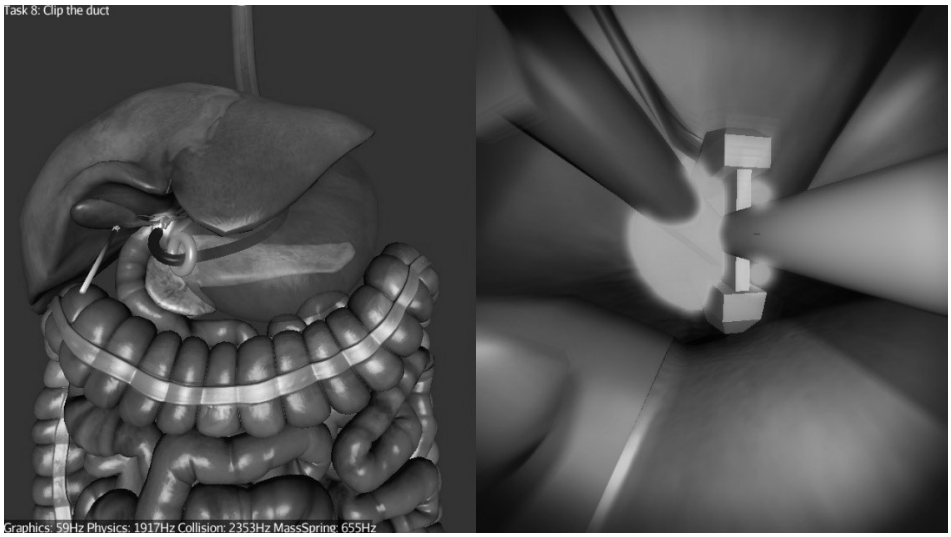


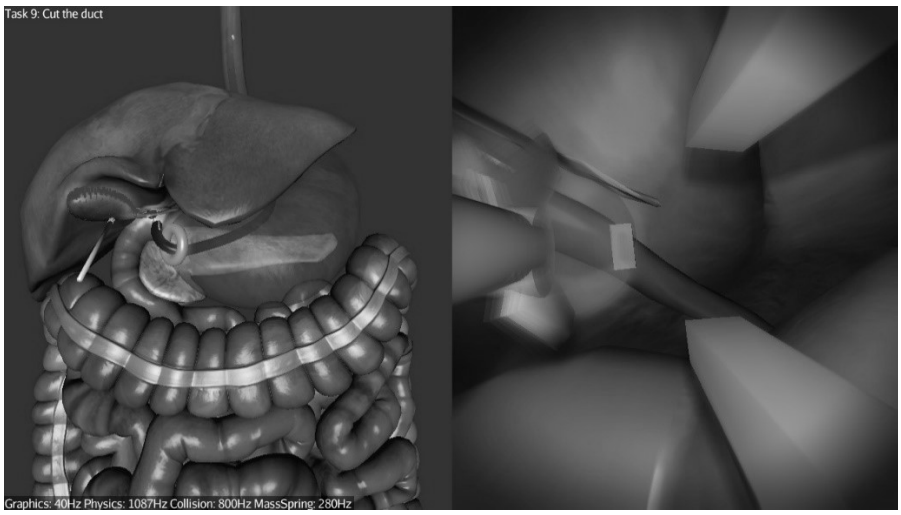
3. Clip and dissect the cystic artery.



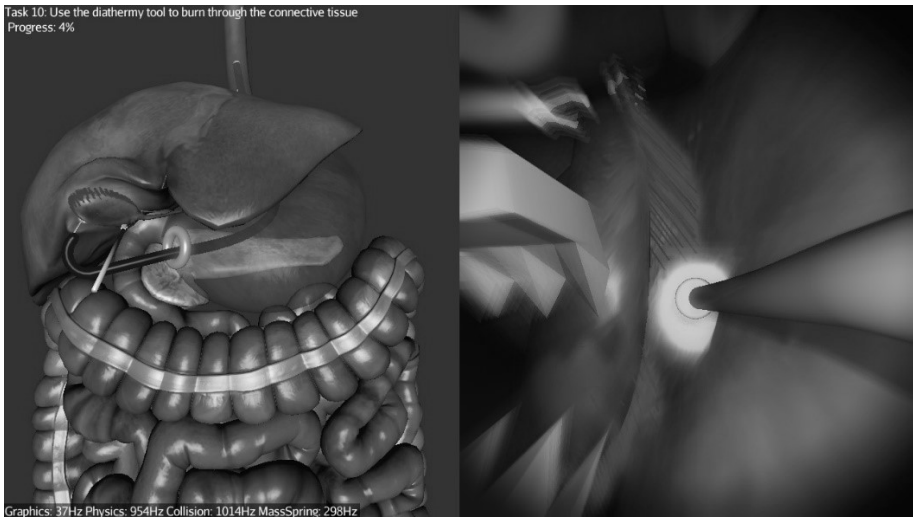


4. Clip and dissect the cystic duct.

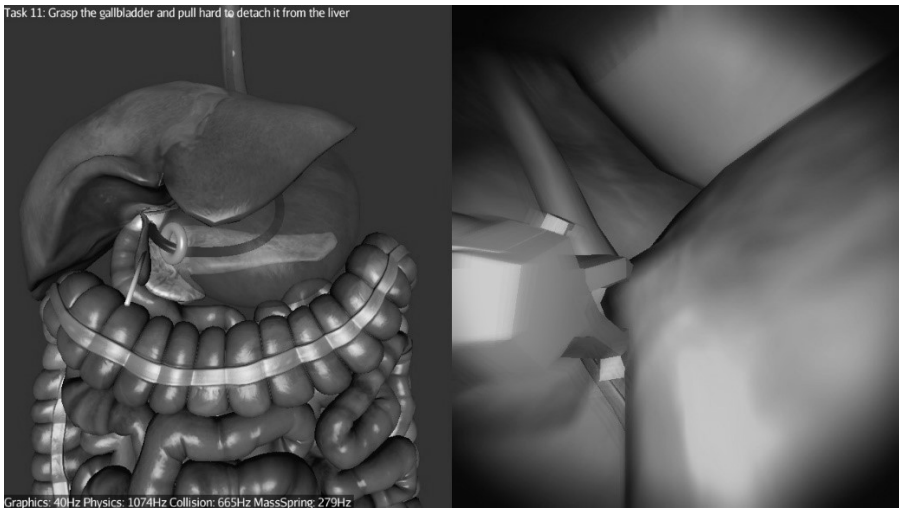




5. Dissect the gallbladder from the liver bed. *N.B As this is a hybrid NOTES cholecystectomy the gallbladder is being retracted by a laparoscopic grasper.* ***You may ask your assistant to increase / decrease the gallbladder retraction.***



6. Remove the gallbladder from the abdominal cavity via the red glowing torus. At this point the simulation will end.



NOTES SIMULATOR PROJECT

Please initial box

1.	I confirm that I have read and understand the participant information sheet dated 3/Jul/2013 version 7.0 for the above study and have had the opportunity to ask questions which have been fully answered.	
2.	I understand that my participation is voluntary and I am free to withdraw and stop taking part at any time without giving a reason.	
3.	I agree to take part in this study.	

_____ _____ _____

Print name of participant **Signature** **Date**

_____ _____ _____

Name of person taking consent **Signature** **Date**

_____ _____ _____

Chief Investigator **Signature** **Date**

Imperial College London

APPENDIX D: VCSIM3 VALIDATION DOCUMENTATION

Validation of VCSim3

A virtual reality simulator for cardiovascular interventions

Research Protocol

Researcher name: Mr. Przemyslaw Korzeniowski

Supervisor: Dr Fernando Bello
Reader in Surgical Graphics & Computing
Division of Surgery, Dept. of Surgery and Cancer

Imperial College London

February 2014

Introduction

Cardiovascular diseases or CVD are the number one cause of death around the world [1]. It is estimated that 17.3 million people died from CVDs in 2008. The key advent of pinhole surgery at the end of the 20th century has enhanced the diagnosis and treatment of many major vascular diseases and has become a vital part of vascular health care today. With this approach, patients suffer much less tissue trauma, which leads to faster recovery and reduced treatment costs since they can usually be treated as day surgery cases [2]. Endovascular clinicians require extensive training and practise because endovascular procedures demand dexterity in handling the delicate guide wire and catheter tools and good hand-eye coordination.

Endovascular clinicians are largely trained using the traditional apprenticeship model where the trainee or apprentice learns first through observation, and then by gradually assisting and performing surgical procedures themselves under the direct supervision of the senior clinician. Several factors such as the increasing costs of time in the operating room [3] have resulted in a need for alternative out-of-the OR training methods such as virtual reality (VR) simulators [4]. Virtual reality simulation has been shown to be a safe and effective training. These simulators have the advantage of being adaptable to simulate different anatomies, as well as having haptic feedback that helps to recreate the feeling of handling the tools through the sense of touch. We have therefore developed a virtual reality cardiovascular simulator - *VCSim*, which allows operators to develop skills in a safe environment and provides them with objective feedback on their performance.

The aim of this study is to establish face, content and construct validity for our simulator in performing cardiovascular interventions. We intend to achieve this by conducting a study that involves gathering the opinions from medical practitioners through an online questionnaire and an experimental study using a prototype of the simulator. The only ethical considerations for the study are voluntary participation, data confidentiality, anonymity and use of the gathered data. This study does not involve patients. It requires only the consensual participation of medical practitioners. The identities of the participants will be anonymised and their responses and any other gathered data will be kept confidential at all times in accordance with the Data Protection Act.

References

[1] Global status report on noncommunicable diseases 2010. Geneva, World Health Organization, 2011.

[2] Benefits of Minimally Invasive Surgery, University of Chicago Webpage, accessed 17 Oct 2013. http://www.uchospitals.edu/specialties/minisurgery/benefits/

[3] The financial impact of teaching surgical residents in the operating room, Bridges, M. ,Diamond, D.L., American Journal of Surgery, 1999 Jan, 177(1), 28-32.

[4] Simulation in Surgical Education, S. de Montbrun, H.MacRae. Clin Colon Rectal Surg,25(3), September 2012, 156-165.

Study Aim & Objectives

The aim of this study is to establish face, content and construct validity for our simulator in performing cardiovascular interventions. The main focus will be put on the behaviour of virtual instruments – catheters and guidewires. Specifically, we want to investigate if the instruments stretch, bend, twist and interact with heart vessels in a realistic way, appropriate for training.

Additionally, we want to examine other aspects of the simulator such as:

- Visual output
- Contrast flow propagation
- Cardiac motion
- Balloon inflation
- Stent deployment
- Haptic feedback

Study Design

We aim to recruit 20-30 participants meeting the entry criteria. Participants' operative experience, demographic data, interest in virtual reality simulation and experience of videogames will be recorded with an online questionnaire.

The experimental apparatus consists of a physical, force feedback human-computer-interface (the haptic device) and a real-time software simulation (the simulation). The complete experimental set-up is presented in Fig. 1 and Fig. 2 in the Appendix.

The simulation software during each update step:

- receives operators movements from the haptic device
- calculates the motion of the virtual catheter and guidewire
- processes the interactions of the catheter and guidewire with the vessels

- measures and stores the performance metrics
- calculates and sends the force feedback back to the haptic device

The software constantly tracks and stores all the movements of the haptic device and of the virtual instruments. The software also stores the metrics such as procedure completion time and applied forces.

Simulation

All participants will be required to complete 5 cardiovascular interventions. Specifically, to navigate the catheter and guidewire from the femoral artery into the heart coronaries, localize the stenosis and deploy a stent. The participants' performance will be analyzed in relation to all their operative attempts and an average will be taken. Prior to performing their first procedure, all participants will be given a technical instruction sheet (*Appendix F*) outlining the nature of the simulation. The aim of this sheet is to give a brief overview of the equipment, tasks and factors, which will differ to real life owing to the limitations of performing the procedure in a simulated setting. After reading the instruction sheet, participants will be given a maximum of 2 minutes to familiarize themselves how to operate the instruments. Prior to commencing their first recorded procedure, participants will be given the opportunity to ask questions relating to the practicalities of the simulation, but will not be allowed to request any technical advice as to how best to perform the procedure. No time limit will be set for the recorded procedures.

Participant data

Participants' operative experience, demographic data, interest in virtual reality simulation and experience of videogames will be recorded with an online questionnaire (https://www.surveymonkey.net/s/vcsim, *printed copy in Appendix A*).In addition, we may also video record the procedure (computer screen and instruments manipulation). If participant wish to withdraw from the study all data related to their participation will be permanently deleted.

Face validity

Face validity will be evaluated by asking participants to complete the aforementioned questionnaire after completing all procedures. The questionnaire will assess the behavior of instruments, graphical appearance, difficulty of the procedure and overall realism.

Content validity

Content validity will be evaluated by asking participants to complete the aforementioned questionnaire after completing all procedures. The questionnaire will assess the adequacy of the

simulated tasks and perceived utility of the simulator as a training tool for cardiovascular interventions.

Construct validity

Construct validity will be evaluated by comparing operative performance metrics of participants.

Intended outcome

It is hoped that the study will lead to the face, content and construct validation of the *VCSim* simulator prototype.

Participant Entry Requirement

Inclusion Criteria

The subjects in this study are also known as endovascular clinicians. They are medical professionals that have been trained or are still in training as interventional radiologists, vascular surgeons and interventional cardiologists. From discussions with several endovascular clinicians, it is recognised that a clinician in training would need to perform a minimum of 300 procedures as the main operating clinician, either with or without senior supervision, in order to gain proficiency. Given that in the UK trainees perform an average of 20 procedures per week, our inclusion criteria is that subjects must have performed endovascular procedures for at least one year.

Exclusion Criteria

Subject has not performed endovascular procedures for a minimum of one year.

Withdrawal Criteria

This is a non-intervention study and there are no consequences for early withdrawal. The subject or participant may withdraw consent at any point.

Data Management

Simulation

Each participant will be assigned a unique identification code. Their data will be carefully anonymised to remove all personal identifiers except for their professional background, position and years of experience. This data will be stored on a secure Imperial College server. There will be a journal/notebook linking this unique code with the individual in case there is a need to go back and look at the personal details again or to exclude a participant's data. The journals will be stored in a locked filing cabinet in a secure office by the data custodian. The appointed data custodian will be Dr Fernando Bello (the study Chief Investigator).

Online questionnaire

The online questionnaire does not require the participant to provide any details that reveal their personal identity other than their medical background and years of professional experience. It is therefore not possible to link the responses of the questionnaire to a specific participant. This ensures the anonymity of participants in the study. Questionnaire responses will be stored online in a secure online account (www.surveymonkey.com).

Adverse Events

This study is not a clinical trial of an investigational medicinal product or medical device. It involves no drugs and no novel procedures. It involves qualitative observations and recordings. Therefore no adverse events are expected.

Assessment and Follow Up

There will be no clinical intervention and therefore no follow-up intervention required.

Data Analysis

To ensure the external validity of the study, we aim to recruit a minimum of 20 participants to complete the online questionnaire. All data will remain anonymous. Data will be analyzed using a statistics package (Statistical Package for the Social Sciences (SPSS) version 18.0.1). Descriptive statistics and frequencies will be calculated with appropriate methods according to the type of data. Significance will be set at $p<0.05$.

Regulatory Issues

Ethics

The main ethical considerations refer to voluntary participation, data confidentiality, anonymity and use of the gathered data. To address participation, an information sheet will be provided to ensure participants are adequately informed. The participants will be assured that the data collected will be confidential, remain anonymous and be used solely for the purposes of this study. The collection and handling of the data will be in accordance to the Data Protection Act.

The research will not involve work done under the Animals (Scientific Procedures) 1986 Act. The research will not involve the use of genetically modified tissue. The project will not involve the use of post-operative, post mortem material or access to confidential patient information.

This protocol will be submitted to the ICREC for ethical approval because this work does not involve NHS patients.

Consent

All participants will be required to sign a written consent form prior to the commencement of the study and will be free to withdraw from the study at any point.

Funding

The development of the VCSim prototype was funded by the EPSRC. Current testing and validation is funded through Health Education England and ICHT.

Study Management

The day-to-day management of the study will be co-ordinated by the researcher Mr Przemyslaw Korzeniowski under the supervision of the Chief Investigator, Dr Fernando Bello.

Publication Policy

Only anonymised data will be used in publication. It is anticipated that resources acquired through the study may be included in presentations at conferences and publications in peer-reviewed journals.

Appendix

A. Online Questionnaire Form
B. Information Sheet for Participants
C. Consent Form for Participants
D. Recruitment Email for Participants
F. Technical Instruction Sheet for Participants

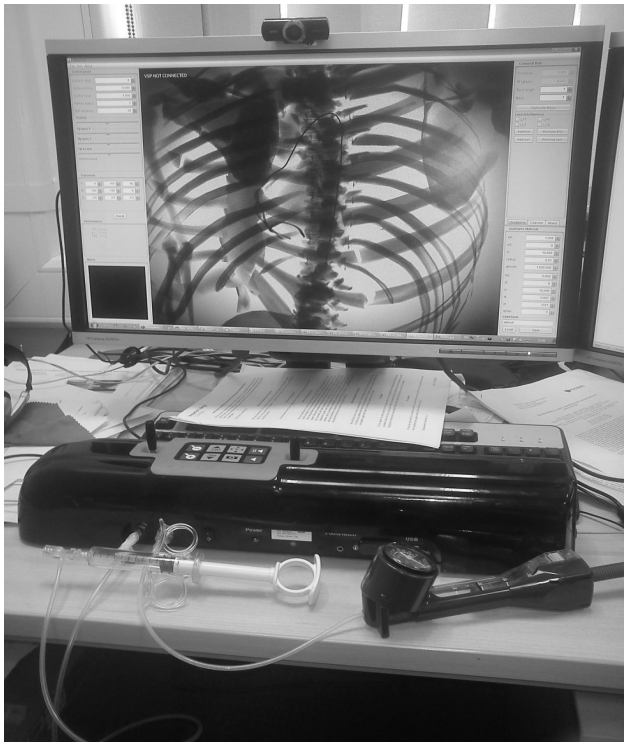


Figure 0.1: A complete experimental set-up. On the desk, the haptic device (black box) with a syringe and a balloon inflation device connected. On the computer screen, a running simulation software.

Figure 0.2: A zoom in at the real instruments (a guidewire inside a catheter) inserted into the haptic device (VSP).

VCSIM Validation

Demographics

***1. Postgraduate year of training - PGY (please only count years with >50% clinical practice).**

PGY

***2. Gender**

- Male
- Female

***3. How often do you play videogames?**

- Very frequently
- Frequently
- Sometimes
- Infrequently
- Very infrequently

***4. Speciality**

VCSIM Validation

Experience

***6. How many of the cardiovascular procedures have you completed INDEPENDENTLY:**

in humans

on virtual reality simulators

on physical simulators

***7. Please rate the following...**

	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
I am interested in the concept of virtual reality simulation in surgery in general	○	○	○	○	○
I believe virtual reality simulation could prove useful in surgical training curricula	○	○	○	○	○

Additiona comments (optional)

VCSIM Validation

Face Validation

***8. Please rate the following statements which pertain to instrument's realism compared to real life...**

	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
The lengths of catheter/guidewire were realistic	○	○	○	○	○
The catheter/guidewire bending behavior was realistic	○	○	○	○	○
The catheter/guidewire twisting behavior was realistic	○	○	○	○	○
The catheter/guidewire stretching behavior was realistic	○	○	○	○	○
The catheter/guidewire tip behavior was realistic	○	○	○	○	○
The catheter/guidewire body (shaft) behavior was realistic	○	○	○	○	○
The interactions between catheter and guidewire were realistic	○	○	○	○	○
The interactions between catheter/guidewire and heart vessels were realistic	○	○	○	○	○
The delay (latency) between physical manipulation and visual reaction was realistic	○	○	○	○	○
The haptic feedback felt realistic	○	○	○	○	○
Overall, the catheter/guidewire behavior felt realistic	○	○	○	○	○

Additional comments

***9. Please rate the following statements which pertain to remaining aspects of the simulator's realism compared to real life...**

	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
The contrast medium injection felt realistic	○	○	○	○	○
The balloon inflation felt realistic	○	○	○	○	○
The stent deployment felt realistic	○	○	○	○	○
The contrast medium propagation was visually realistic	○	○	○	○	○
The balloon inflation was visually realistic	○	○	○	○	○
The stent deployment was visually realistic	○	○	○	○	○
The cardiac motion was visually realistic	○	○	○	○	○
The visualization as a whole looked realistic	○	○	○	○	○
The difficulty of the simulated procedure was realistic	○	○	○	○	○
Overall the simulator was realistic	○	○	○	○	○

Additional comments

Content Validation

***11. Please rate the following statements which pertain to how usfeul the simulator is as a training tool for cardiovascular interventons...**

	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
The catheter/guidewire behavior is sufficient to make it a useful training tool for cardiovascular interventions	◯	◯	◯	◯	◯
The remaining functionality provided by the simulator is sufficient to make it a useful training tool for cardiovascular interventions	◯	◯	◯	◯	◯
Overall the simulator is a useful training tool for cardiovascular interventions	◯	◯	◯	◯	◯
I would recommend the simulator to others	◯	◯	◯	◯	◯

Additional comments

PARTICIPANT INFORMATION SHEET FOR VCSIM 3.0 VALIDATION STUDY

We are running a research study and we wish to enter you as a participant. We would like you to take the time to read this sheet, which explains the research. Ask us if there is anything that is not clear or if you would like more information. Please take your time to decide whether or not you wish to participate.

Thank you for reading this.

What is the purpose of the research?

We have developed a virtual reality simulator for the cardiovascular intervention. The purpose of the research is to determine how useful the simulator is as a training tool.

Why have I been chosen?

You have been chosen as your surgical experience satisfies our participant selection criteria.

What are you asking of me?

We are asking you to perform five cardiovascular interventions on our virtual reality simulator. Specifically, to navigate the catheter and guidewire from the femoral artery into the heart coronaries, localize the stenosis and deploy a stent. We expect that the whole study shouldn't take more than 20 minutes. Your performance will be recorded and kept securely in an electronic file. We may also video record the procedure (operative screen and instruments manipulations). After you have completed all procedures, we require you to complete an online electronic questionnaire. This will take approximately 5 minutes and will ask questions relating to your experience with the simulator. Your responses will be kept securely on an electronic file. All the data gathered will be anonymised and used only for the purpose of the study.

What will happen if I take part?

We will ask you to sign a consent form. The form breaks down the consent process into parts. This is to help you understand what you are agreeing to. You should read the consent form carefully and, if you agree, sign each part. If you do not agree then do not sign.

If I agree?

If you agree to participate in the study then this is recorded on the consent form. We will discuss suitable times for you to participate in the experiments.

If I refuse?

If you do not wish to continue with the study then we will record this on the consent form. We will not contact you again about this research study.

What about data, confidentiality and privacy?

We plan to keep the data related to this research for 10 years. The data will be kept secure by the Data Custodian. Access to the data will only be for research staff. Data will be anonymised. If you wish to see your recorded input after providing them, this will be permitted, however you will not be allowed to view other participants' data. If you wish to withdraw from the study all data related to your participation will be permanently deleted.

What are the benefits of taking part?

You will gain a better understanding of the potential role that a virtual reality simulator could play in training and assessment of the operative techniques. You contribution may also lead to more realistic VR simulators in the future.

What if something goes wrong?

If there is a technical fault with the experimental setup during the tests, which prevents you from completing it, the fault and the stage in the test at which it happened will be recorded before restarting the setup to the stage at which it stopped. If needed, a separate session will be arranged to complete the tests.

What will happen to the results of the research?

We hope that the results will be published in a scientific journal. The data will be anonymised so the readers of the journal will not know who the operators were.

Who is organising and funding the research?

This research is organised by the Faculty of Medicine at Imperial College London. The development of the VCSim prototype was funded by the EPSRC. Current testing and validation is funded through Health Education England and ICHT.

Who has reviewed this study?

This study was given a favourable ethical opinion for conduct by the Imperial College Research Ethics Committee (ICREC). The ICREC was founded in 2006 to review studies which need ethical consideration, but which fall outside the remit of NHS Ethics Committees. The Committee comprises of 4 lay members and 4 members of Imperial College.

Contact for Further Information

For further information please contact Dr Fernando Bello. He can be contacted on the following telephone number: 0203312 1788

Thank you for reading this information sheet.

IF YOU DECIDE TO CONSENT TO THIS STUDY A COPY OF THIS SHEET AND A SIGNED CONSENT FORM WILL BE GIVEN TO YOU TO KEEP.

VCSIM 3.0 SIMULATOR VALIDATION STUDY

CONSENT FORM

Please initial box

4.	I confirm that I have read and understand the participant information sheet dated 26 February 2014 version 1.2 for the above study and have had the opportunity to ask questions which have been fully answered.	
5.	I understand that my participation is voluntary and I am free to withdraw and stop taking part at any time without giving a reason.	
6.	I agree to take part in this study.	

_____	_____	_____
Print name of participant	**Signature**	**Date**
_____	_____	_____
Name of person taking consent	**Signature**	**Date**
_____	_____	_____
Chief Investigator	**Signature**	**Date**

Recruitment Email

Dear Dr Xxx,

My name is Przemyslaw Korzeniowski and I am a Research Assistant in the Department of Surgery and Cancer at Imperial College London. My work focuses on developing virtual reality simulators for cardiovascular interventions. Primarily, I am looking into modelling the behaviour of catheters and guidewires i.e., how these instruments bend, twist, stretch and how they interact with the heart vessels.

I am currently conducting a face and construct validation of the simulator prototype. The study requires navigating the instruments 5 times from the femoral artery into the heart coronaries. The software will record performance metrics such as completion times and used forces. We may also video record movements of your hands. The whole study shouldn't take more than 20 minutes. Afterwards, you will be asked to fill a short (5 minutes) questionnaire assessing the behaviour of the instruments.

All the gathered data will be anonymised and used only for the purpose of the study.

I would be very grateful if you could participate. Your involvement will not only be extremely helpful to my work, but it will lead to more realistic VR simulators in the future.

With kind regards,

Przemyslaw Korzeniowski

Validation of VCSim 3.0
Technical Instruction Sheet

A virtual reality simulator for cardiovascular interventions

As part of this study you have been requested to perform 5 simulated cardiovascular interventioons on a virtual reality simulator. This sheet has been designed to provide you with an introduction to the equipment (Fig. 1) that you will be using, as well as an overview of the nature of the simulation.

EQUIPMENT

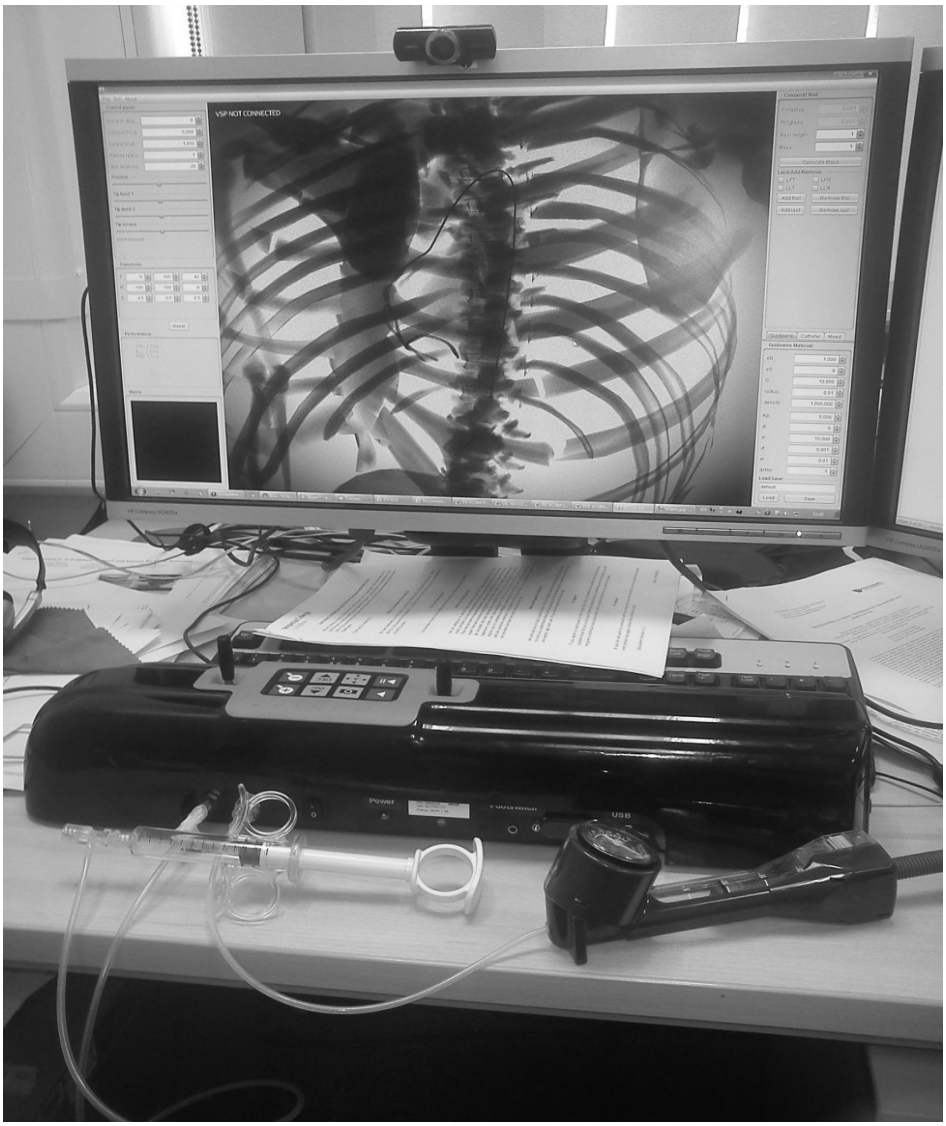


Figure 0.3: A complete experimental set-up. On the desk, the haptic device (black box) with a syringe and a balloon inflation device connected. On the computer screen, a running simulation software.

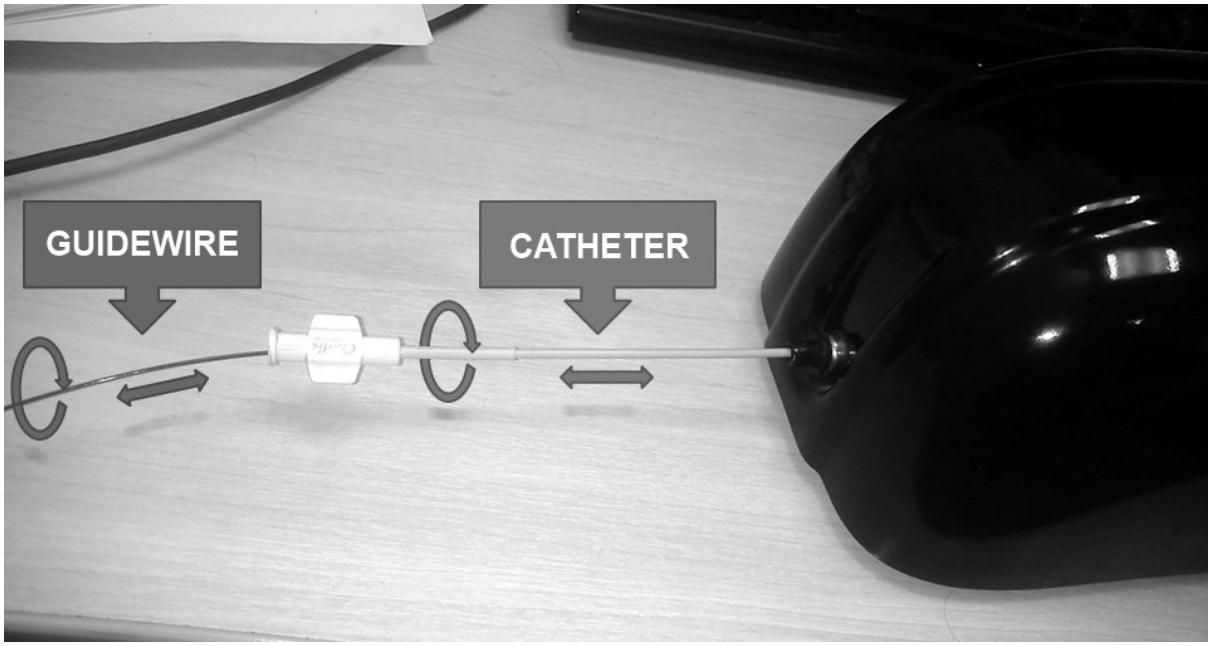


Figure 0.4: A zoom in at the real instruments (a guidewire inside a catheter) inserted into the haptic device (VSP).

The Fig.2 show the haptic device that you will be using.

Insert / withdraw the cather/guidwire further into the machine to advance / withdraw the instruments. *N.B. The length of the instruments is less than that of real ones. During the simulation, you may occasionally exceed the available length. If this happens, the screen will fade out, the simulation will be paused and you will be asked to withdraw the instruments. Once the instruments have been withdrawn to an appropriate position, the simulation will re-commence with the instruments in an unaltered position on the screen*

Twist the instruments clockwise / anticlockwise to torque them.

QUESTIONS

You may now ask the investigator any questions you have. Please note that although the investigator can answer questions relating to the practicalities of the simulation they cannot offer any technical advice as to how best to perform the procedure.

BASIC SIMULATION

Prior to commencing the procedural, you will be given a maximum of 2 minutes to practice basic instrument navigation in a test environment. Your performance during this basic simulation will not be analysed

PROCEDURAL SIMULATION

The procedural simulation will start with the instruments in the aorta. There is no time limit. You should take the same care as you would if you were operating on a human.